Building the Complexity of Rational Cubics for Designing the Objects

by

Azhar Sayeed

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

December, 1996

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

REDUCING THE COMPLEXITY OF RATIONAL CUBICS FOR DESIGNING THE OBJECTS

BY

AZHAR SAYEED

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

DECEMBER 1996

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN, SAUDI ARABIA

COLLEGE OF GRADUATE STUDIES

This thesis, written by

AZHAR SAYEED

under the direction of his Thesis Advisor and approved by his Thesis Committee, has been presented to and accepted by the Dean of the College of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Thesis Committee

Dr. Muhammad Sarfraz (Chairman)

Dr. Jarallah S. Al – Ghamdi (Member)

Dr. Muhammad S. Al – Mulhem (Member)

Department Chairman

Dean, College of Graduate Studies

1/31/97

Date

كلية الدراسات العليا

COLLEGE OF GRADUATE STUDIES

Acknowledgments

I would like to acknowledge King Fahd University of Petroleum and Minerals for all the support extended during this research.

Thanks are due to my thesis committee chairman, Dr. Muhammad Sarfraz for his encouragement, support, and cooperation. I would also like to thank my thesis committee members, Dr. Jarallah S. Al-Ghamdi and Dr. Muhammad S. Al-Mulhem for their helpful comments and suggestions.

Thanks are due to the KFUPM community for providing a friendly environment.

Contents

List of Tables

List of Figures

THESIS ABSTRACT

Name: AZHAR SAYEED

Title: REDUCING THE COMPLEXITY OF RATIONAL CUBICS FOR DESIGNING THE OBJECTS

Degree: MASTER OF SCIENCE

Major Field: INFORMATION & COMPUTER SCIENCE

Date of Degree: DECEMBER 1996

Computer Graphics mainly deals with the pictorial representation of objects on computer. Now-a-days graphical user interfaces are widely used on many operating systems. Graphical User Interfaces (GUI's) provide a natural way of interaction with the computers.

Splines are used for generating curves and surfaces. For smooth generation of curves and surfaces spline subdivisions are done. Subdivisions are useful for displaying approximation spline curves, until the control graph approximates the curve path. Control point coordinates then can be plotted as curve position. Spline subdivision/conversion can be easily applied for other spline representation. Interactive curve design is typically accomplished through the manipulation of control polygon. To achieve, the desired shape of a curve or surface, spline subdivision is applied. This research is oriented towards the conversion of rational cubic B-splines into rational quadratic B-splines.

The process of conversion (rescue realization) is taken from Pratt. The rational cubic curve segment is divided into two halves, with two different conics (rational quadratics) representing each curve segment. This division into rational quadratics provides an economical alternative to the rational cubics. Shape parameter are associated with each interval to tighten the rational splines on intervals and/or at the control points. The process of conversion resulted in a C^1 shape preserving scheme for quadratic B-splines.

Keywords: B-Splines, Computer Graphics, Division, Rescue Realization, Rational Cubic B-Splines, Rational Quadratic B-Splines, Time Complexity.

King Fahd University of Petroleum and Minerals, Dhahran.
December 1996

ملخص الرسالة

الاســـم : أزهر سيد

عنوان الرسالة : تقليل تعقيد التكعيبات النسبية لتصميم الكائنات

الدرجـــة : ماجستير علوم

التخصـــص : علم الحاسب و المعلومات

تاريخ الدرجة : ديسمبر ١٩٩٦

الرسم بالحاسب الآلي يتعامل بشكل رئيسي مع التمثيل التصويري للكائنات في الحاسب ، وهذه الأيام تستخدم بشكل واسع واجهات التطبيق بالرسم في العديد من أنظمة التشغيل . و هذه الواجهات تقدم الطريقة الطبيعية للتعامل مع الحاسب . و الشرائح تستخدم لتشكيل المنحنيات و الأسطح . و التشكيل المنتظم للمنحنيات و الأسطح تستخدم تقسيمات الشرائح . و هذه التقسيمات تفيد في عرض منحنيات الشرائح التقريبية إلى أن يقرب رسم التحكم طريق المنحنى . و من ثم فإن إحداثيات نقاط التحكم ترسم كموقع للمنحنى . و تقسيمات الشرائح ز تحويلها سمكن أن تطبق بسهولة على تمثيلات الشرائح الأخرى . و تصميم المنحنيات التحادثي عادة ما ينجز بالتعامل مع مضلع التحكم . و للوصول إلى الشكل المطلوب للمنحنى أو السطح تستخدم تقسيمات الشرائح . وهذا البحث يقوم بتحويل الشرائح التكعيبية النسبية إلى الشرائح التربيعية النسبية.

عملية التحويل مأخوذة من برات . و قطعة المنحنى التكعيبية النسبية بقسم إلى قسمين بمخروطين مختلفين (تر بيعات نسبية) لتمثيل كل قطعة من المنحنى . و هذا التقسيم التربيعي النسبي يقدم بديلاً أوفر من التكعيبات النسبية . و عوامل الشكل ترتبط مع كل فترة لتضييق الشرائح النسبية على الفترات في نقاط التحكم . و عملية التحويل أنتجت خطة المحافظة على الشكل C^1 لتر بيعات شرائح B .

المفاتيح: شرائح B ، الرسم بالحاسب ، التقسيم ، شرائح B التكعيبية النسبية ، شرائح B التربيعية النسبية ، تعقيد الوقت ، تمثيل الإنقاذ .

جامعة الملك فهد للبترول و المعادن

ديسمبر ١٩٩٦

Chapter 1

Introduction

Graphic representation constitutes one of the basic sign systems conceived by the human mind for the purpose of storing, understanding, and communicating essential information. As a "language" for the eye, graphics benefits from the ubiquitous properties of visual perception. As a monosemic system, it forms the rational part of the world of images.

Graphics owes its special significance to its double function as a storage mechanism and a research instrument. A rational and efficient tool when the properties of visual perception are competently utilized, graphics is one of the major "language" applicable processing. Electronic displays, such as the cathode ray tube, open up an unlimited future to graphics [1].

Computer Graphics is concerned with representation of distinct objects through computers. Computer Graphics are used in a wide variety of applications such as science, engineering, medicine, architecture, entertainment, education, training etc. Now-a-days graphical user interfaces are widely used on many operating systems. These *Graphical User Interfaces (GUI's)* provides a natural way of interaction with the computers.

Computer Graphics has changed the way people perceive and use computers. It is one of the fastest growing field of computer science, due to its numerous applica-

tions, such as modelling real world objects, performing graphical simulations, etc. Development of pictures in computer graphics are achieved through employment of curves and surfaces [2]. Curves and Surfaces are used in the construction of quite distinct objects like car bodies, ship hulls, airplane fuselages and wings, propeller blades, shoe insoles, pots, etc.

This chapter gives an overview of the spline. First, the discussion considers, what a spline is and why they are useful. The following section gives a detail description of different kinds of splines. The chapter closes with a description of where the remaining chapters take us.

Computer graphics applications involves smooth curves and surfaces to be generated. As many real world objects, computer aided-design (CAD), high quality character fonts, data plots and artists sketches are smooth in representation. The path or movement of a camera, object in the animation sequence is also smooth.

The need to represent curves and surfaces arises in two cases:

- **Modelling existing objects (a car, a face, a mountain)**

  In the process of designing or modelling an existing objects, the mathematical description of the object may not be available. In such case the coordinates of the object are used to model it.

  Due to limited storage in computer, it is not preferred. The object is usually approximated using pieces of planes, spheres and any other shapes that are easy to describe it mathematically. The required points of the model be close to the corresponding points of the object.

- **Modelling "from scratch"**

  When there is no information available to model the object. The object is

created in the modelling process, to match its representation as desired.

The user can create the object interactively, develop the mathematical representation for it, and give an approximate description to be filled in by some program. In CAD, the computer representation is used to generate physical realizations of the abstractly designed object.

Descriptive geometry was used to solve the above problems. Surface was defined by a set of curves, usually plane sections with some characteristic feature lines. The information was sufficient to manufacture templates. Master models (wooden) are developed using the templates. Stamps and dies are obtained from the master models by means of copy milling.

In the late fifties, milling machines were derived by numerical control. The machining instructions were generated by a computer program. The information is stored in a computer compatible form. Thereby translating an existing surface definition into a computerized model by the following way.

- A given shape contour is digitized, there by translating a continuous outline to a discrete (digitized) version (*This information can be obtained from a physical model itself or from the drawing board).*

- The discrete model is then modelled through the use of a *Spline* construct (*Advantage of it is in the speed of capture, portability within and between computer systems*).

- The described outline can be easily rotated, scaled and translated through applying a simple and appropriate transformation matrix.

1.1 Need for B-splines

Graphics is one of the most fundamental and difficult activities associated with visualization. This lead to the development of many different approaches and methods. One of the most often used is of *spline construct.*

Mathematical descriptions are of various forms. Application of a specific mathematical description (linear, quadratic, cubic or even higher) depends upon the type of data fit required. Line segments are used to model a curve outline, requiring line segments to meet reasonable number of design specifications. Depending how adjacent arc's are join together (the type of continuity between them), the decision of using linear, quadratic, cubic or even higher ordered modeler is used.

Higher degree of continuity or smoothness are offered by increasing the order of the mathematical description. This will result in outlines and joints appearing "smoother" in shape as desired by the user. Smoothness is achieved at the cost of higher computation, necessary for the mathematical description. Therefore, for the purpose of less computation, curve descriptions above cubic order are not preferred for modelling the outlines of the shape. The concept of *B-splines* are used and applied in it.

1.2 History of B-splines

N. Lobachevsky [3] investigated B-splines in the early nineteenth century, it was constructed as convolutions of certain probability distribution functions. I.J. Schoenberg [4] in 1946, started the modern mathematical theory of spline approximation, using B-splines for statistical data smoothing.

B-splines are the powerful tool for Computer Aided Geometric Design (CAGD) and are found in the existing CAD/CAM systems. They are the basis for the space of the n^{th} degree splines of continuity class C^{n-1}. A B-spline is a non-negative n^{th} degree spline, non-zero only on n+1 intervals. B-splines form a partition of unity, such that, they sum to one. Curves generated by summing control points multiplied by the B-splines have some very desirable shape properties, including the local *convex hull* and *variation diminishing* property.

Applications of the splines are in the design of curve and surface shapes, digitize drawings for computer storage, specify animation paths for the objects or the camera in a scene. Real world applications include design of automobile bodies, aircraft and spacecraft surfaces and ship hulls.

A curve is made up of a number of curve segments joined together with some continuity constraint. Curve segments exhibit certain continuity properties. When a curve is made up of curve segments, continuity property changes depending on the way segments have been joined. To ensure a smooth transition from one curve segment to another, continuity conditions are imposed at the connection points, such as geometric and parametric continuity.

1.3 Continuity Conditions of B-splines

Parametric Continuity: A parametric continuity C^n is defined as follows: if the direction and magnitude of curve segment $\frac{d^n}{dt^n}[Q(t)]$ are equal at the join point between segments then the curve exhibits C^n continuity.

Geometric Continuity: Simply joining two segments together at a common end point implies G^0 geometric continuity. If tangent vectors to each curve segment at

the common end point match to within a constant (direction equal, magnitudes not equal) then the curve has G^1 continuity and also the parametric first derivatives are proportional at the intersection of two successive segments. If the first and second parametric derivatives of the two curve segments (sections) are proportional at their boundary, then the curve has G^2 continuity, curvatures of two curve sections match at their joining position.

1.4 Spline Specification

There are three equivalent methods for specification of a particular spline representation [5].

- Imposing boundary conditions on the spline.
- A matrix characterization of spline.
- By a set of blending functions (basis functions) that determine how specified geometric constraints on the curve are combined to calculate positions along the curve path.

The definition of spline is given below:

Definition 1 *An Mth degree spline is a piecewise polynomial of degree M that has continuity of derivatives of order M-1 at each knot.*

In computer graphics, the term spline curve refers to any composite curve formed with polynomial sections satisfying specified continuity conditions at the boundary of the pieces, and a spline surface is described with two sets of orthogonal spline curve.

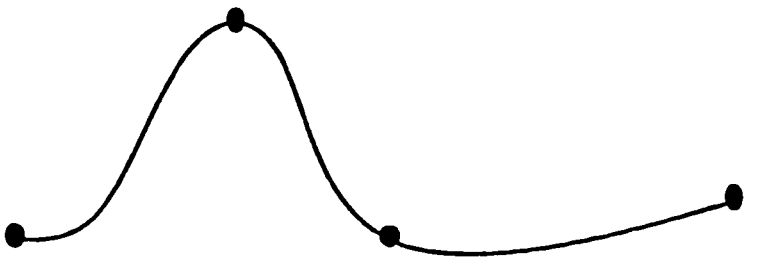

Figure 1.1: Interpolatory Spline

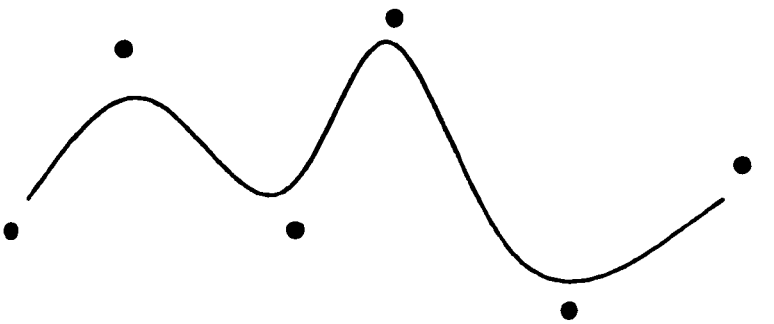

Figure 1.2: Approximatory Spline

1.5 Classification of Splines

A spline curve is specified by giving a set of coordinate positions, called control points, indicating the shape of a curve. Spline curve is defined, modified and manipulated with operations on the control points. Control points are then fitted with piecewise continuous parametric polynomial functions in one of two ways.

The polynomial section fitted, so that the curve passes through each control point, the resulting curve is said to interpolate the set of control points. Such type of splines are known as *Interpolatory splines.* (see figure.1.1.) Interpolatory splines are used to digitize drawings, specify animation paths etc.

When polynomials are fitted to the general control-point path without necessarily passing through any control point, the resulting curve is said to approximate the

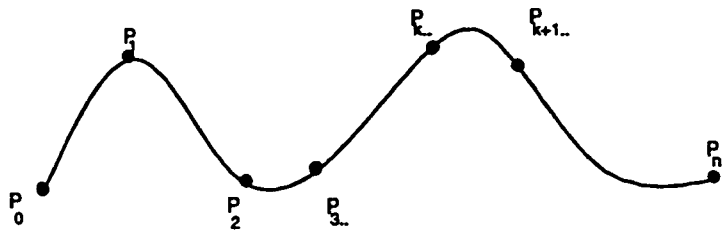

Figure 1.3: A piecewise continuous natural cubic spline interpolation of n+1 control points

set of control points, such type of splines are known as *approximatory splines,* (see figure.1.2.) Approximation curves are used as design tools to structure object surfaces. Cubic spline Interpolation methods are often used to set up paths for object motions or to provide a representation for an existing object or drawing, and also to design object shapes.

Cubic polynomials offer a reasonable compromise between flexibility and speed of computation, compared to higher order polynomials. They require less computation, memory and are more stable. Compared to lower order polynomials, cubic splines are more flexible for modeling arbitrary curve shapes.

1.6 Interpolatory Splines

1.6.1 Natural Cubic Spline

The mathematical equivalent of earlier strips (splines) are natural cubic spline, (see figure.1.3.), these are C^2 continuous cubic polynomials that interpolate (pass through) the control points. It has one more degree of continuity than is inherent in the Hermite and Bezier forms.

These splines have no "local control". The shape of the curve is specified by

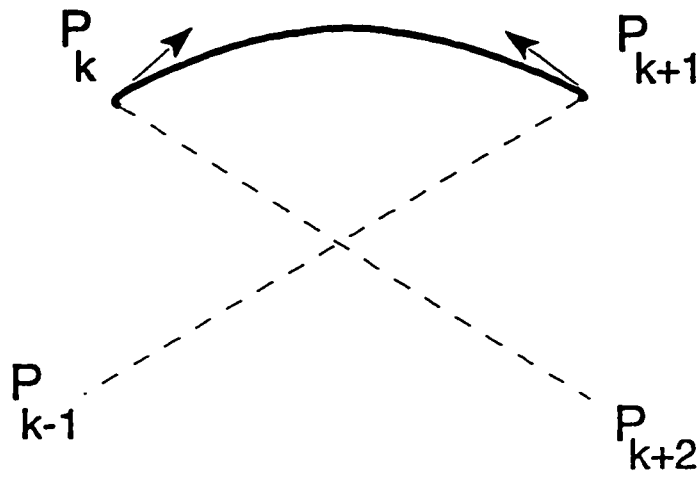

Figure 1.4: Cardinal Spline

the position of control points, it is difficult to restructure part of the curve without specifying new set of control points.

1.6.2 Hermite Spline

It is an interpolatory piecewise cubic polynomial with a specified tangent at each control point. These splines exhibit "local control" as each curve section is only dependent on its end point constraints.

Cardinal and Kochanek-Bartels splines are variations on the Hermite splines.

1.6.3 Cardinal Spline

These are specified with 4 consecutive control points, (see figure 1.4.) The middle two control points are the section end points, and the other two points are used in the calculation of endpoint slopes. *Catmull-Rom Splines/Overhauser splines* interpolate the middle two control points, the tangent at those points is parallel to the line

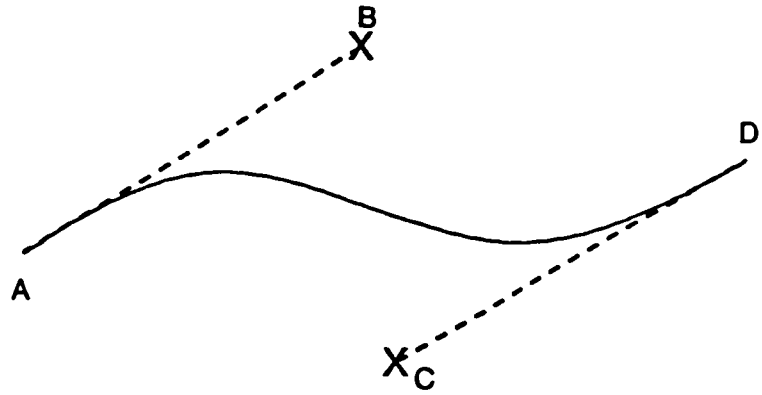

Figure 1.5: A Two Dimensional Bezier Curve.

between the previous control point and the next control point.

1.6.4 Kochanek-Bartels Spline

are extensions of cardinal splines, additional parameters are introduced to provide further flexibility in adjusting the shape of the curve. Parametric derivatives may not be continuous across section boundaries. These were designed to model animation paths.

1.7 Approximatory Splines

1.7.1 Bezier Curves

The Bezier form of the cubic polynomial curve segment, named after Pierre Bezier, interpolates the two end control points and approximates the other, see figure 1.5.

The Bezier curve is a polynomial of degree one less than the number of control points. This makes calculation of polynomial functions of higher degree expensive in terms of computation. Curve follows the shape of the control point polygon and

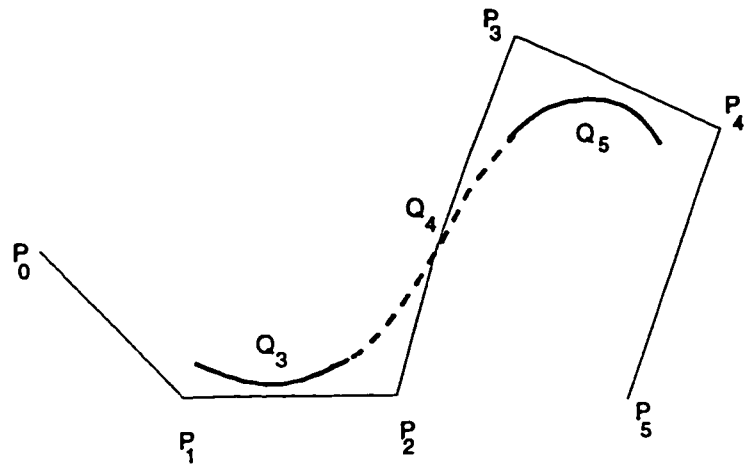

Figure 1.6: A three segment B-spline cubic curve.

is constrained within the convex hull formed by the control points. Bezier curves do not allow for local control of the curve shape, repositioning any one of the control point affects the entire curve.

Cubic Bezier curve is only a special case of a set of Bernstein-Bezier [1] polynomial curves. Cubics are used most often because they are reasonably simple, but are still sufficiently flexible for interactive design work.

The curve doesn't exhibits variation diminishing property (i.e., the curve does not oscillate about any straight line more often than the control point polygon). Curve is invariant under affine transformation.

[1]The polynomials which are weights are called Bernstein polynomials.

1.7.2 Uniform Nonrational B-splines

B-spline curves were used to overcome the drawbacks of Bezier curves, such as non-localness and the relationship between the degree of the curve and the number of control points. B-splines are more complex than Bezier splines.

A general expression for the calculation of coordinate positions along a B-spline curve in a blending function formulation is of the form

$$P(t) = \sum_{k=0}^{n} P_k B_{k,d}(t), \quad t_{min} \leq t \leq t_{max}, \quad 2 \leq d \leq n+1 \tag{1.1}$$

P_k are an input set of n+1 control points, and the B-spline blending functions $B_{k,d}$ are polynomials of degree d-1

B-splines consist of curve segments whose polynomial coefficients depend on just a few control points, exhibiting local control, thereby moving a control point affects only a small part of the curve. The time to compute the coefficients are greatly reduced, have same continuity as natural splines.

Cubic B-splines approximate a series of n+1 control points $P_0, P_1, \ldots, P_n$, n ≥ 3, with a curve consisting of n-2 cubic polynomial curve segments $Q_3, Q_4, \ldots, Q_n$. Such cubic curves might be defined each on its own domain $0 \leq t \leq 1$, the parameter t is adjusted so that the domains for the various curve segments are sequential. Thus the parameter range on which Q_i is defined is $t_i \leq t \leq t_{i+1}$, for $3 \leq i \leq$ m. In case of n=3, there is a single curve segment Q_3 that is defined on the interval $t_3 \leq t \leq t_4$ by four control points; P_0, P_1, P_2, P_3. (see figure.1.6)

For each i$\geq$4, there is a join point or knot between Q_{i-1} and Q_i at the parameter value t_i, the parameter value at such a point is called a knot value. The initial and

final points at t_3 and t_{n+1} are also called knots, so that there is a total of n-1 knots. A closed B-spline curve is created by repeating the set of beginning points. If the knots are spaced at equal intervals of the parameter t; then these are referred to as uniform. The "B" stands for basis, since the splines can be represented as weighted sums of polynomial basis function.

Each of the n-2 curve segments of a B-spline curve is defined by four of the n+1 control points. In particular curve segment Q_i is defined by points $P_{i-3}, P_{i-2}, P_{i-1}$ and P_i, curve segment Q_i begins somewhere near point P_{i-2} and ends somewhere near point P_{i-1}.

B-spline blending functions are everywhere nonnegative and sum to unity, so the curve segment Q_i is constrained to the convex hull of its four control points. As each curve segment is defined by four control points, each control point influences four curve segments. This is the *local control* property of B-splines. If $P_{i-2} = P_{i-1}$, curve is pulled closer to this point, as curve segment Q_i is defined by just three different points, and the point $P_{i-2} = P_{i-1}$ is weighted twice.

1.7.3 Nonuniform, Nonrational B-splines

The parameter interval t between sucessive knot values need not be uniform. The nonuniform knot value sequence means that the blending functions are no longer the same for each interval, rather it varies from curve segment to curve segment.

These curve segments have several advantages over uniform B-splines. First, continuity at selected join points can be reduced from C^2 to C^1 to C^0 to no. If the continuity is reduced to C^0, then the curve interpolates a control point, without the undesirable effect of uniform B-splines, where the curve segments on either side of

the interpolated control points are straight lines. Also, the starting and ending point can be easily interpolated exactly. The increase generality of nonuniform B-splines require a slightly different notation than that used for uniform B-splines.

As the cubic spline is a piecewise continuous curve made up of cubic polynomials, approximating the control points P_0 through P_n. The knot value sequence is a nondecreasing sequence of knot values t_0 through t_{n+4} (four extra knots than the number of control points). As the smallest number of control points is four, smallest knot sequence has eight knot values and the curve is defined over the parameter interval from t_3 to t_4.

The only restriction on the knot sequence is that it be nondecreasing, this allows successive knot values to be equal, the number of identical parameter values is called the multiplicity of the knot.

Curve segment Q_i is defined by the control point $P_{i-3}, P_{i-2}, P_{i-1}, P_i$ and by the blending functions $B_{i-3,4}(t)$, $B_{i-2,4}(t)$,$B_{i-1,4}(t)$,$B_{i,4}(t)$ as the weighted sum.

$$Q_i(t) = \sum P_{i-3} B_{i-3,4}(t)$$

The curve is not defined outside the interval t_3 through t_{n+1}, when $t_i = t_{i+1}$ (a multiple knot), curve segment Q_i is a single point. There is no single set of blending functions, as there was for the other types of splines.

The functions depend on the intervals between knot values and are defined recursively in terms of lower order blending functions.

$B_{i,j}(t)$ is the jth order blending function for weighting control point P_i.

The recursive method for cubic B-splines is

$$B_{i,1}(t) = \begin{cases} 1, \text{ for } t_i \leq t \leq t_{i+1} \\ 0, \text{ otherwise} \end{cases}$$

$$B_{i,2}(t) = \frac{t - t_i}{t_{i+1} - t_i} B_{i,1}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} B_{i+1,1}(t)$$

$$B_{i,3}(t) = \frac{t - t_i}{t_{i+2} - t_i} B_{i,2}(t) + \frac{t_{i+3} - t}{t_{i+3} - t_{i+1}} B_{i+1,2}(t)$$

$$B_{i,4}(t) = \frac{t - t_i}{t_{i+3} - t_i} B_{i,3}(t) + \frac{t_{i+4} - t}{t_{i+4} - t_{i+1}} B_{i+1,3}(t)$$

Blending functions are nonnegative and sum to one. So nonuniform B-spline segments lie within the convex hull of their four control points.

Increasing knot multiplicity has two effects.

- First each knot value will lie within the convex hull of the points.
- Multiple knots will reduce parametric continuity from C^2 to C^1 (one extra knot;multiple 2), C^1 to C^0 (two extra knot;multiple 3), C^0 to no continuity for 3 (multiple 4),

1.7.4 Nonuniform, Rational Cubic Polynomial Curve Segments

Rational cubic curve segments are ratios of polynomials.

$$x(t) = \frac{X(t)}{W(t)};\ y(t) = \frac{Y(t)}{W(t)};\ z(t) = \frac{Z(t)}{W(t)};$$

X(t), Y(t), Z(t) and W(t) are cubic polynomial curves whose control points are defined in homogenous coordinates.

A curve in homogenous space is given as Q(t)=[X(t), Y(t), Z(t), W(t)], moving from homogenous space to 3-space involves dividing by w(t). Any nonrational curve can be transformed to a rational curve by adding w(t)=1 as a fourth element. These polynomials in a rational curve can be Bezier, Hermite or any other type. When they are B-splines, we have nonuniform rational B-splines, sometimes called NURBS.

Rational curves are important for the following two reasons.

- They are invariant under rotation, scaling, translation and perspective transformations of the control points, where as nonrational curves are invariant under only rotation, scaling and translation.
- Rational splines define precisely any of the cubic section.

1.8 Subdividing Curves

When a series of connected curve segments are created to approximate a given shape for designing purpose, by manipulation of the control points, and still the desired shape is not achieved; it is probably due to less control points to achieve the desired effect.

There are two ways to increase the number of control points. One is the process of *degree elevation:* The degree of spline is increased from 3 to 4 or more. This adjustment is sometimes necessary, especially if higher orders of continuity are needed, but is generally undesirable because of the additional inflection points allowed in a single curve and the additional computation time needed to evaluate the curve.

The second or more useful way to increase the number of control points is to subdivide one or more of the curve segments into two segments. For instance,

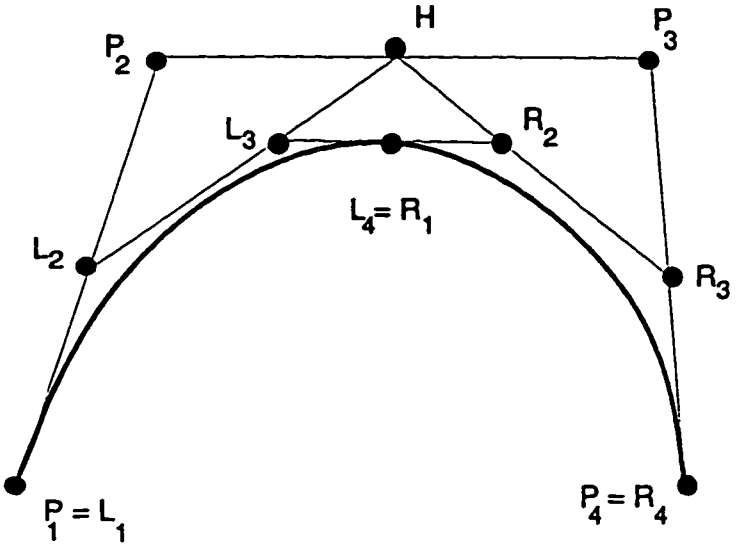

Figure 1.7: Subdividing a Cubic Bezier curve at t=1/2.

a Bezier curve segment with its four control points can be subdivided into two segments with a total of seven control points; i.e., the two segments share a common control point. The two new segments exactly match the one original segment until any of the control points are actually moved; then one or both of the new segments no longer match the original. Another reason for subdividing is to display a particular curve or surface.

Given a Bezier curve Q(t), see figure.1.7. defined by points P_1, P_2, P_3 and P_4, we find a left curve defined by points L_1, L_2, L_3 and L_4, and a right curve defined by points R_1, R_2, R_3 and R_4, such that the left curve is coincident with Q on the interval $0 \leq t < \frac{1}{2}$ and the right curve is coincident with Q on the interval $\frac{1}{2} \leq t < 1$. The subdivision can be accomplished using a geometric construction technique developed by de-Casteljau [6] to evaluate a Bezier curve for any value of t.

The point on the curve for a parameter value of t is found by drawing the

construction line L_2H so that it divides P_1P_2 and P_2P_3 in the ratio of t:1-t, HR_3 so that it similarly divides P_2P_3 and P_3P_4, and L_3R_2 to likewise divide L_2H and HR_3. The point L_4(which is also R_1) divides L_3R_2 by the same ratio and gives the point Q(t). The figure.1.7 shows the case for t=1/2.

The new control points L_i and R_i is a weighted sum of the points P_i, with the weights positive and summing to 1. Thus, each of the new control points is in the convex hull of the set of control points. Therefore, the new control points are no longer farther from the curve Q(t) than are the original control points, and in general are closer than the original points. This *variation-diminishing* property is true of all splines that have the convex hull property.

Dividing the Bezier curve at t=$\frac{1}{2}$ often gives the interactive user the control needed, but it might be better to allow the user to indicate a point on the curve at which a split is to occur. Given such a point, it is easy to find the approximate corresponding value of t. The splitting proceeds as described, except that the construction lines are divided in the ratio of t:1-t. Another subdivision method specified in [5] is the recursive spline subdivision, it consists of repeatedly dividing a given curve section in half, thereby increasing the number of control points at each step.

Subdivisions are useful for displaying approximation spline curves, until the control graph approximates the curve path. Control point coordinates then can be plotted as curve position. Another importance is to generate more control points to shape the curve. Line adjustments can be made to small sections of the curve.

Spline subdivision can be easily applied to other spline representations with the following sequence of operations.

1. Convert the spline representation in use to a Bezier representation.

2. Apply Bezier subdivision algorithm.

3. Convert the Bezier representation back to the original spline representation.

In this thesis, a method for division of rational cubic splines developed in [7] is demonstrated theoretically and practically. This division is composed of two rational quadratics for each piece of rational cubic spline. Later a recursive subdivision algorithm can be considered for practical implementations.

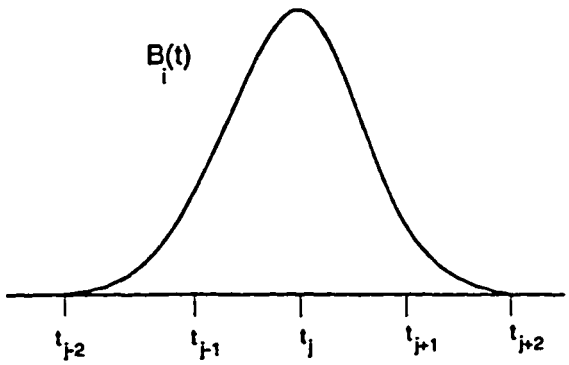

Figure 1.8: Cubic B-spline

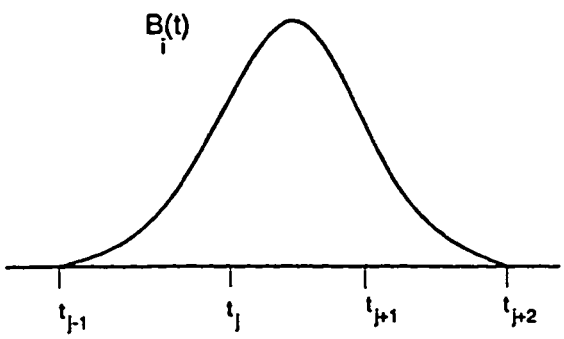

Figure 1.9: Quadratic B-spline

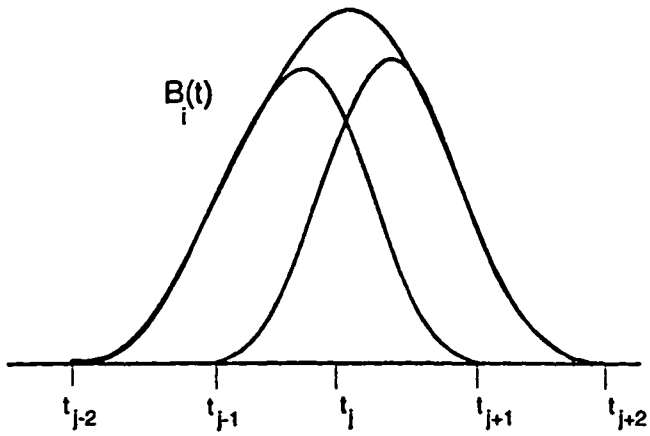

Figure 1.10: Approximating Cubic B-spline with Quadratic B-splines.

1.9 Problem Definition

The problem addressed by this work is to reduce the complexity of the rational cubics [7] for designing the objects. This process involves the following steps.

- Converting the Rational Cubic B-spline developed in [7] into Rational Quadratic (Conic) B-splines [8].
- Implementation of Rational Cubic [7] and Rational Quadratic(Conic) B-splines [8].
- Comparison of Rational Cubic and Rational Quadratic(Conic) B-splines [8].

These developments are of significance from both a theoretical and practical point of view. As the conversion results in achieving, the reasonable results with less computation from rational quadratic (conic) B-splines. It shows an improvement in terms of *Time Efficiency*. The cubic B-spline is of the form as shown in figure 1.8. The quadratic B-spline is of the form as in figure 1.9. By approximating the cubic B-spline using quadratic B-splines as shown in figure 1.10. The design curve to be generated will be of the approximating form for cubic B-spline using the quadratic B-splines.

1.10 Thesis Organization

Chapter 2 presents other work related to subdividing of spline, concentrating on other research in subdividing/splitting of cubic B-splines, with respect to control polygon. It also revisits the rational cubic B-splines in [7] and rational quadratic B-splines in [2]. Prepares a framework for the conversion of cubics into conics, similar

to that developed in [2].

Chapter 3 illustrates how rational quadratics (conics) rescue rational cubic B-splines. Rational Cubic B-splines are converted into rational quadratic B-splines. A formal method of the work is presented, giving a mathematical description of both rational cubic B-splines and rational quadratic B-splines.

Chapter 4 compares the rational cubic B-splines with rational quadratic (conics) B-splines developed. Presents conclusions, gives the direction for future research utilizing the method mentioned.

Chapter 2

Previous and Related Work

This chapter examines previous research done on the subdivision of cubic splines. A number of other researchers have looked at the question of subdivision and splitting of cubic B-splines (Interpolatory and Approximatory).

2.1 Overview

Subdivision has many applications in computer graphics. A single subdivision operation will divide a curve or surface. Successive (recursive) subdivisions can be used to generate linear approximations of curves or surfaces which can then be used to develop efficient plotting and display algorithms [9, 10, 11]. Subdivision yields accurate and robust intersection algorithms when combined with convex hull properties.

An example is the intersection algorithm for two Bezier surfaces. Catmull developed the first subdivision algorithm for computer display of curved surfaces [12]. His algorithm subdivides a parametrically defined general form cubic curve using a kind of difference equation for obtaining the midpoint of the curve. The method

was later extended to general form of bicubic parametric surfaces. Catmull used his algorithm to recursively subdivide a bicubic surface patch until the resulting subpatch was the size of a pixel of the raster display on which it has to be rendered.

Subdividing a Bezier curve at the midpoint in parametric space yields two segments of Bezier curves. The two sets of control points (control polygons) which define the two resulting Bezier curve segments can be obtained using the well-known de-Casteljau algorithm [6, 13]. The method of dividing Bezier surface patches is the extension of the curve method where four new control nets are produced defining four corresponding subpatches.

Any form of bicubic parametric surface can be subdivided by either converting it into power basis form using Catmull's algorithm or converting it into Bezier form and then using the algorithm for subdividing Bezier surfaces. In[9], an efficient algorithm for subdividing a linear Coons surface[1] patch without first converting it to the general form or the Bezier form is illustrated. The boundary curves are assumed to be cubic Bezier curves.

When dealing with CAGD techniques for constructing interpolant, the methods used are Bezier, B-splines, β-splines and rational splines. All these methods use a "control polygon" and allow fast computation of the resulting function or curve using subdivision algorithms like de-Casteljau's. These methods preserve the shape of the control polygon and are local. When interpolation is required at some points they become global.

Dyne et al.[14] introduced the idea of *interpolatory subdivision* which amounts to provide a fast local algorithm to compute a "smooth" interpolant. The result

[1] A Coons surface patch is defined by four boundary curves and blending functions for the interior of the patch.

is not necessarily shape preserving. Much of the work done by Chaiken [15], Doo and Sabin [16] dealt with the problem of producing curves by limits of polygons, but they are neither shape preserving nor interpolatory. In [17], a shape preserving interpolatory subdivision method is presented. The method preserves the convexity and is local, the resulting interpolant is a C^1 function known only through its computation algorithm.

A subdivision preserving the weak convexity of Bezier nets is shown in [18], i.e., the weak convexity condition of Bezier nets defined on a base triangle T is preserved on any subtriangles inside T.

In each of the subdivision (recursive) methods mentioned above, a curve segment (bicubic) is divided into two segments (each cubic) with respect to the control polygon. In this thesis, a rational cubic curve segment is divided into two rational quadratic curve segments. A rational cubic curve segment is of degree three, after division of it into two quadratic curve segments each will be of degree two. Later, any of the recursive division techniques mentioned above can be used to develop an algorithm for subdivision.

2.2 Cubics and Conics

The following section revisits the rational Cubic B-splines developed in [7] and the rational quadratic B-splines in [2].

2.2.1 Overview

Polylines and polygons are first degree, piecewise linear approximations to curves and surfaces, respectively. Unless the curves or surfaces being approximated are also piecwise linear, large number of end point coordinates must be created and stored to achieve reasonable accuracy [2].

Interactive manipulations of the data to approximate a given shape is tedious as many points have to be positioned precisely. The general approach is to use function that are higher degree than are the linear functions. They still approximate only the desired shapes, but use less storage and offer easier interactive manipulation than do linear functions.

2.3 Cubics

Cubic polynomials are most often used as lower-degree polynomials give too little flexibility in controlling the shape of the curve, and higher-degree polynomials can introduce unwanted wiggles and also require more computation time [6].

In computer aided geometric design (CAGD) a common problem is to interactively design a parametric curve or surface represented by a given set of control points. The shape of a curve or surface is generally changed by either manipulating the control point or by modifying the shape parameter associated with the surface

fitting method.

The surface splines [19, 20], are approximate/freeform tensor product surfaces and can be modified by either changing the control points or by using different shape parameters.

There is a lack of local control by the shape parameters in the tensor product methods. It can be achieved, however by the use of rational cubics (NURBS) [13, 21] but this method also has some limitations. To generate a rational B-spline surface that has interval tension parameters to control the shape of the system, a bivariate B-spline product is used [7]. C^2 variable weights are used. Such that the spline surface is C^2 continuous.

2.3.1 Rational Cubic Spline

In the following section, a freeform rational cubic tension spline developed by Sarfraz [7] is revisited. This rational cubic B-spline is taken as the basis for conversion into rational quadratic (conic) B-spline.

2.3.2 C^1 Rational Cubic Hermite Interpolant

According to Sarfraz in [7], "A piecewise rational cubic Hermite parametric function $\phi_j(t)\epsilon[t_0, t_n]$, with parameter r_i, i=0,...,n-1 is defined for $t\epsilon[t_i, t_{i+1}]$, i=0,...,n-1, and is given as

$$\phi_j(t) = \frac{(1-\theta)^3 + \theta(1-\theta)^2(r_i F_i + h_i D_i) + \theta^2(1-\theta)(r_i F_{i+1} - h_i D_{i+1} + \theta^3 F_{i+1})}{1 + (r_i - 3)\theta(1-\theta)} \tag{2.1}$$

where $r_i \geq 0$, are used as tension parameters to control the shape of the curve. The case for r_i=3, i=0,...,n-1, is of cubic Hermite interpolation and the restriction $r_i > -1$ in it ensures a positive denominator as mentioned in equation 2.1. F_i is the values at knots t_i, i=0,...,n where $t_0 < t_1 < ... < t_n$, and the $D_i \epsilon R^N$, i=0,...,n representing the first derivative values defined at the knots.

The above function $\phi_j(t)$ developed has the Hermite interpolation properties:

$$\phi_j(t_i) = F_i \text{ and } \phi_j(t_i)^{(1)} = D_i, \text{ for i=0,...,n.}$$

For $r_i \neq 0$, equation 2.1 is written in the following form:

$$\phi_j(t) = R_0(\theta; r_i)F_i + R_1(\theta; r_i)V_i + R_2(\theta; r_i)W_i + R_3(\theta; r_i)F_{i+1} \tag{2.2}$$

where

$$V_i = F_i + h_i D_i / r_i, \; W_i = F_{i+1} - h_i D_{i+1} / r_i,$$

and $R_j(\theta; r_i)$, j=0,1,2,3 are appropriately defined as rational functions with

$$\sum_{j=0}^{3} R_j(\theta; r_i) = 1. \tag{2.3}$$

Moreover, these functions are rational Bernstein-Bezier weight functions which are non-negative for $r_i > 0$. Thus in R^N, $N > 1$ and for $r_i > 0$ the convex hull property holds i.e., the curve segment lies in the convex hull of the control points F_i, V_i, W_i, F_{i+1}. Moreover, the variation diminishing property also holds of the rational cubic i.e the curve segment crosses any (hyper) plane of dimensions N-1 no more times than it crosses the control polygon joining F_i, V_i, W_i, F_{i+1}.

The rational cubic of equation 2.1 can be expressed as:

$$\phi_j(t) = l_i(t) + e_i(t; r_i),$$

where

$$l_i(t) = (1-\theta)F_i + \theta F_{i+1}$$

$$e_i(t; r_i) = \frac{h_i\theta(1-\theta)[(\delta_i - D_i)(\theta - 1) + (\delta_i - D_{i+1})\theta]}{1 + (r_i - 3)\theta(1-\theta)}$$

and

$$\delta_i = \frac{F_i - F_{i+1}}{h_i}$$

This immediately lead to Interval tension property i.e., for given fixed (or bounded) D_i, D_{i+1}, the rational cubic Hermite interpolant of equation 2.1 converges uniformly to the linear interpolant on $[t_i, t_{i+1}]$ as $r_i \to \infty$.

In the following section, a C^2 rational spline interpolant is constructed. This requires the knowledge of the second derivative of equation 2.1 which, after some simplification, is given by

$$\phi_j(t)^{(2)} = \frac{2\alpha_i\theta^3 + \beta_i\theta^2(1-\theta) + \gamma_i\theta(1-\theta)^2 + \delta_i(1-\theta)^3}{h_i 1 + (r_i - 3)\theta(1-\theta)^3} \tag{2.4}$$

where

$$\alpha_i = r_i(D_{i+1} - \delta_i) - D_{i+1} + D_i$$

$$\beta_i = 3(D_{i+1} - \delta_i)$$

$$\gamma_i = 3(\delta_i - D_i)$$

$$\delta_i = r_i(\delta_i - D_i) - D_{i+1} + D_i.''$$

2.3.3 C^2 Rational Cubic Spline Interpolant

According to Sarfraz as mentioned in [7]," The familiar procedure of allowing the derivative parameters D_i, i=0,...,n to be degrees of freedom which are constrained by the impositions of the C^2 continuity conditions are considered

$$\phi_j(t_{i+})^{(2)} = \phi_j(t_{i-})^{(2)}, i = 1, ..., n-1. \tag{2.5}$$

These C^2 conditions give from equation 2.4, the linear system of consistency equations,

$$h_i D_{i-1} + h_i(r_{i-1} - 1) + h_{i-1}(r_i - 1)D_i + h_{i-1}D_{i+1} = h_i r_{i-1}\delta_{i-1} + h_{i-1}r_i\delta_i, i = 1, ..., n-1. \tag{2.6}$$

For simplicity, assume that D_0 and D_n are given as end conditions (clearly other end conditions are appropriate). Assume also that

$$r_i \geq r > 2, \ \ i = 0, ..., n-1$$

Then the equation 2.6 defines a diagonally dominant, tri-diagonal linear system in the unknowns D_i, i=1,...,n-1. Hence there exists a unique solution which can be easily calculated by use of the tri-diagonal LU decomposition algorithm. Thus a rational cubic spline interpolant can be constructed with tension parameters r_i, i=0,...,n-1, where the special case r_i=3, i=0,...,n-1 is that of cubic spline interpolation".

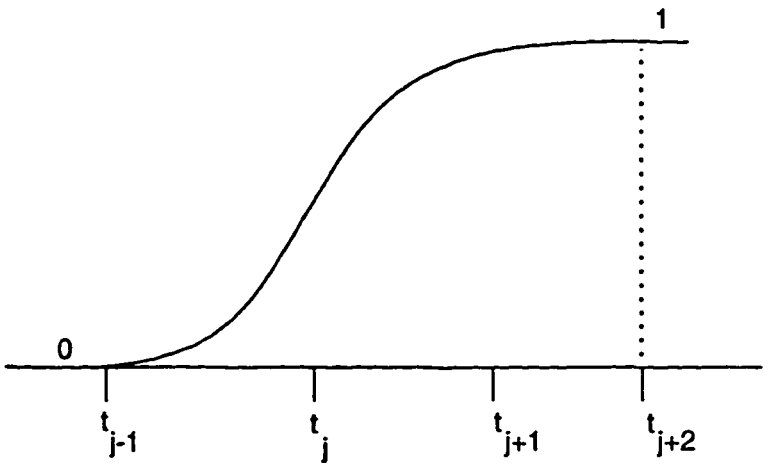

Figure 2.1: Basis function $\phi_j(t)$

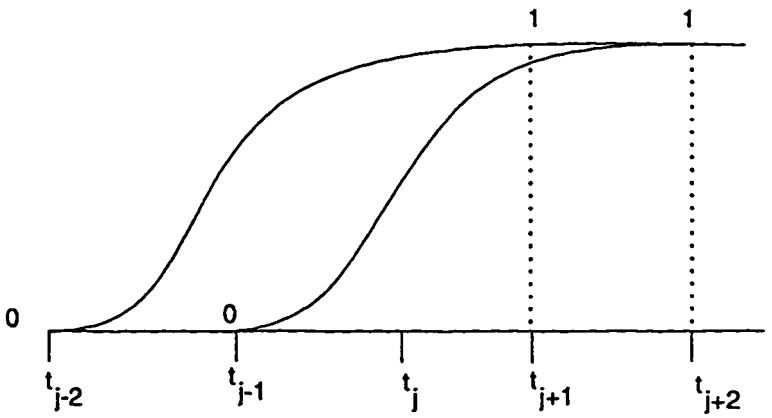

Figure 2.2: The function $\phi_j(t)$ and $\phi_{j+1}(t)$.

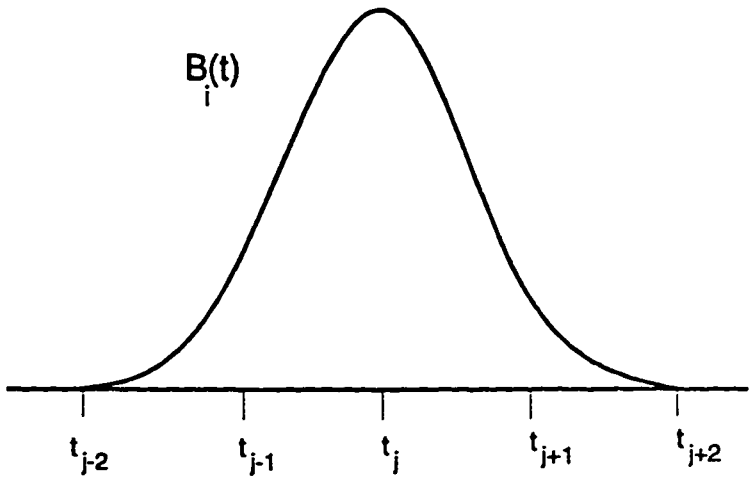

Figure 2.3: Cubic B-spline B_j(t)

2.3.4 Rational Cubic Spline With Tension

A freeform rational cubic tension spline curves in [22] is revisited. As mentioned by J. A. Gregory and M. Sarfraz in [22], "For the purpose of analysis, additional knots are introduced outside the interval $[t_0, t_n]$, defined by $t_{-3} < t_{-2} < t_{-1} < t_0$ and $t_n < t_{n+1} < t_{n+2} < t_{n+3}$.

And the shape parameter r_i defined on the extended partition is as follows.

$$t_{-3} < t_{-2} <, \ldots, t_{n+2} < t_{n+3} \text{ as } r_i \geq r > 2, \text{ for } i = -3, \ldots, n+2,$$

The basis function $\phi_j(t)$ (see figure 2.1) is constructed such that

$$\phi_j(t) = \begin{cases} 0, & for\ t < t_{j-1} \\ 1, & for\ t \geq t_{j+2} \end{cases}$$

(The function ϕ_{j+1}(t) is constructed as in figure 2.2). The local support rational cubic spline, basis $B_j(t)$, j = -1,...,n+1 is defined as in [7] by the difference functions

(see figure 2.3).

$$B_j(t) = \phi_j(t) - \phi_{j+1}(t) \tag{2.7}$$

A local support rational cubic B-spline basis B_j(t), j = -1,...,n+1 is constructed and an explicit representation of it is given as

$$\begin{aligned} B_j(t) &= R_0(\theta; r_i)B_j(t_i) + R_1(\theta; r_i)(B_j(t_i) + h_i B_j^{(1)}(t_i)/r_i) \\ &\quad + R_2(\theta; r_i)(B_j(t_{i+1}) - h_i B_j^{(1)}(t_{i+1})/r_i) + R_3(\theta; r_i)B_j(t_{i+1}) \end{aligned}$$

where

$$B_j(t_i) = B_j^{(1)}(t_i) = 0 \ \ for \ \ i \neq j-1, j, j+1$$

and

$$\begin{aligned} B_j(t_{j-1}) = \mu_{j-1}, \ \ B_j^{(1)}(t_{j-1}) = \hat{\mu}_{j-1}, \\ B_j(t_j) = 1 - \lambda_j - \mu_j, \ \ B_j^{(1)}(t_j) = \hat{\lambda}_j - \hat{\mu}_j \\ B_j(t_{j+1}) = \lambda_{j+1}, \ \ B_j^{(1)}(t_{j+1}) = -\hat{\lambda}_{j+1} \end{aligned} \tag{2.8}$$

with

$$\begin{aligned} \lambda_j = h_j\hat{\lambda}_j/r_j, \mu_j = h_{j-1}\hat{\mu}_j/r_{j-1}, \\ \hat{\lambda}_j = h_j d_{j-1}/c_j, \hat{\mu}_j = h_{j-1}d_{j+1}/c_{j+1}. \end{aligned} \tag{2.9}$$

$$c_j = h_{j-2}d_j(\frac{h_{j-2}}{r_{j-2}} + \frac{h_{j-1}}{r_{j-1}}) + h_j d_{j-1}(\frac{h_{j-1}}{r_{j-1}} + \frac{h_j}{r_j}) + \frac{h_{j-1}d_{j-1}d_j}{r_{j-1}}$$

$$d_j = h_j(r_{j-1} - 2) + h_{j-1}(r_j - 2)$$

$h_i = t_{i+1} - t_i; \theta \equiv \theta(t) = \frac{(t-t_i)}{h_i}$ and $R_k(\theta; r_i)$, k = 0,1,2,3 are defined as.,

$$R_0(\theta; r_i) = \frac{(1-\theta)^3}{Q_0(\theta; r_i)}$$

$$R_1(\theta; r_i) = \frac{r_i\theta(1-\theta)^2}{Q_0(\theta; r_i)}$$

$$R_2(\theta; r_i) = \frac{r_i\theta^2(1-\theta)}{Q_0(\theta; r_i)}$$

$$R_3(\theta; r_i) = \frac{\theta^3}{Q_0(\theta; r_i)}$$

where

$$Q_0(\theta; r_i) = 1 + (r_i - 3)\theta(1-\theta),$$

also

$$Q_0(\theta; r_i) = (1-\theta)^3 + r_i\theta(1-\theta)^2 + r_i\theta^2(1-\theta) + \theta^3$$

The above $R_k(\theta; r_i)$, k=0,1,2,3 are actually the Bernstein-Bezier weight functions.[2] These rational spline functions are such that they satisfy the B-spline properties.

- Local Support Property: $B_j(t) = 0$ for $t \notin (t_{j-2}, t_{j+2})$,

[2]The polynomials which are weights are called Bernstein polynomials.

- Partition of Unity: $\sum_{j=-1}^{n+1} B_j(t) = 1$, for $t\epsilon[t_0, t_n]$,
- Positivity: $B_j(t) \geq 0$ for all t,

The design curve is given by:

$$p(t) = \sum_{j=-1}^{n+1} P_j B_j(t), \quad t\epsilon[t_0, t_n], \tag{2.10}$$

where $P_j \epsilon R^N$ define the control points, was transformed to the piecewise defined rational Bernstein-Bezier representation

$$p(t) = R_0(\theta; r_i)F_i + R_1(\theta; r_i)Q_i + R_2(\theta; r_i)R_i + R_3(\theta; r_i)F_{i+1} \tag{2.11}$$

where

$$\begin{aligned} F_i &= \lambda_i P_{i-1} + (1 - \lambda_i - \mu_i)P_i + \mu_i P_{i+1}, \\ Q_i &= (1 - \alpha_i)P_i + \alpha_i P_{i+1}, \\ R_i &= \beta_i P_i + (1 - \beta_i)P_{i+1}. \end{aligned} \tag{2.12}$$

with

$$\begin{aligned} \alpha_i &= \mu_i + h_i\hat{\mu}/r_i = \hat{\mu}_i(h_{i-1}/r_{i-1} + h_i/r_i) \\ \beta_i &= \lambda_{i+1} + h_i\hat{\lambda}_{i+1}/r_i = \hat{\lambda}_{i+1}(h_i/r_i + h_{i+1}/r_{i+1}). \end{aligned} \tag{2.13}$$

Representing the transformation 2.13 in the matrix notation.

$$\mathrm{X_i} = \mathrm{Y_i Z_i} \tag{2.14}$$

where $X_i = [F_i \ Q_i \ \ R_i \ F_{i+1}]^T$, $Z_i = [P_{i-1} \ \ P_i \ \ P_{i+1} \ \ P_{i+2}]^T$ and

$$Y_i = \begin{bmatrix} \lambda_i & 1-\lambda_i-\mu_i & \mu_i & \\ & 1-\alpha_i & \alpha_i & \\ & \beta_i & 1-\beta_i & \\ & \lambda_{i+1} & 1-\lambda_{i+1}-\mu_{i+1} & \mu_{i+1} \end{bmatrix} \tag{2.15}$$

The transformation to rational Bernstein Bezier form are very conveinient for computational purposes. It therefore leads to the following properties.

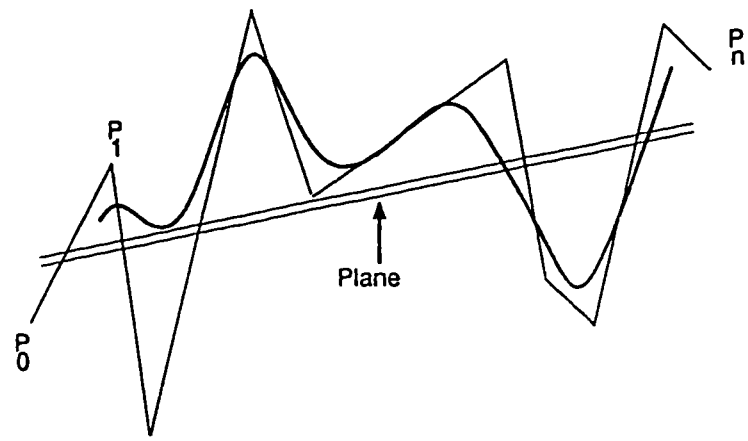

Figure 2.4: Variation Diminishing Property

- Proposition 1. Variation Diminishing Property: The rational B-spline curve p(t) tϵ[t_0, t_n] defined by equation 2.10 crosses the (hyper) plane of dimension N-1 no more times than it crosses the *control polygon* P joining the control points $\{P_j\}_{j=-1}^{=n+1}$(see figure 2.4).

- Proposition 2. Linear B-spline tension property: Let $r_i \geq r > 2$, i = j-2,...j+1 and $||.||$ denote the uniform norm. Then

$$\phi_j(t) = \begin{cases} (t - t_{j-1})/h_{j-1}, & t_{j-1} \leq t < t_j \\ (t_{j+1} - t)/h_j, & t_j \leq t < t_{j+1} \\ 0, \ otherwise \end{cases}$$

is the linear polynomial B-spline.

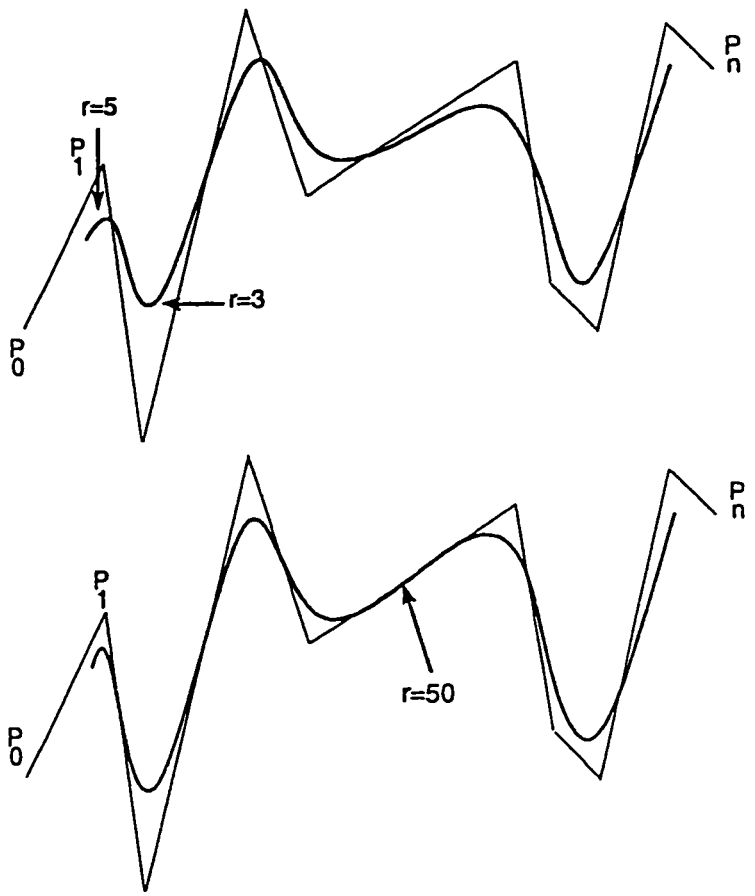

Figure 2.5: Illustration of Global/Local tension property.

- Proposition 3. Global tension property: Let $r_i \geq r > 2$, i=-2,...,n+1, and let P denote the rational B-spline control polygon, defined explicitly on $[t_i, t_{i+1}]$, i = -1,...,n, by

$$P(t) = (1-\theta)P_i + \theta P_{i+1},\ \theta \equiv \theta(t) = (t-t_i)/h_i \tag{2.16}$$

Then the rational B-spline converges uniformly to P on $[t_{-1}, t_{n+1}]$ as r $\rightarrow \infty$.

- Proposition 4. Interval tension property: let

$$Q_k = (1-\mu)P_k + \mu P_{k+1}$$
$$Q_{k+1} = \lambda P_k + (1-\lambda)P_{k+1}$$

denote two distinct points on the line segment of the control polygon joining P_k, P_{k+1} where

$$\lambda = \frac{h_{k+1}/r_{k+1}}{(h_{k-1}/r_{k-1})+(h_{k+1}/r_{k+1})+h_k},$$
$$\mu = \frac{h_{k-1}/r_{k-1}}{(h_{k-1}/r_{k-1})+(h_{k+1}/r_{k+1})+h_k},$$

Q_K is the first point and is before Q_{K+1}, since $\lambda + \mu < 1$. Then the rational B-spline converges uniformly to Q on $[t_K, t_{K+1}]$ as r $\rightarrow \infty$, where Q(t)=(1-θ)Q_k+θQ_{K+1}, $\theta(t) = (t-t_k)/h_k$";

Figure 2.5 illustrate the interval and global tension behaviour of the curves. As the value of r_k and r_{k+1} increases the resulting curve segment approaches the line segment P_k, P_{k+1}.

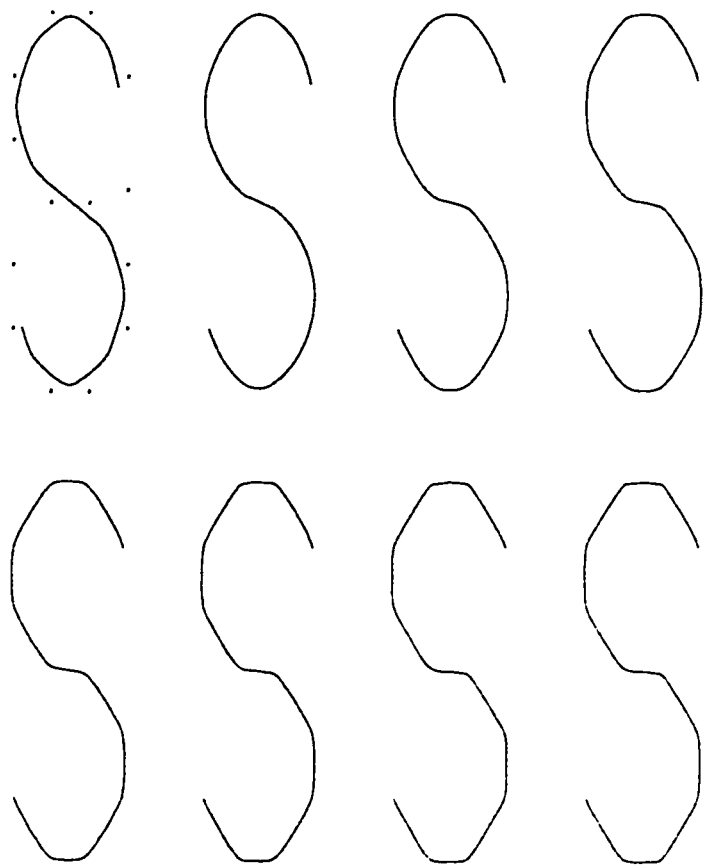

Figure 2.6: Freeform rational cubic spline curves drawn for shape parameter range 2 to 9 (left to right).

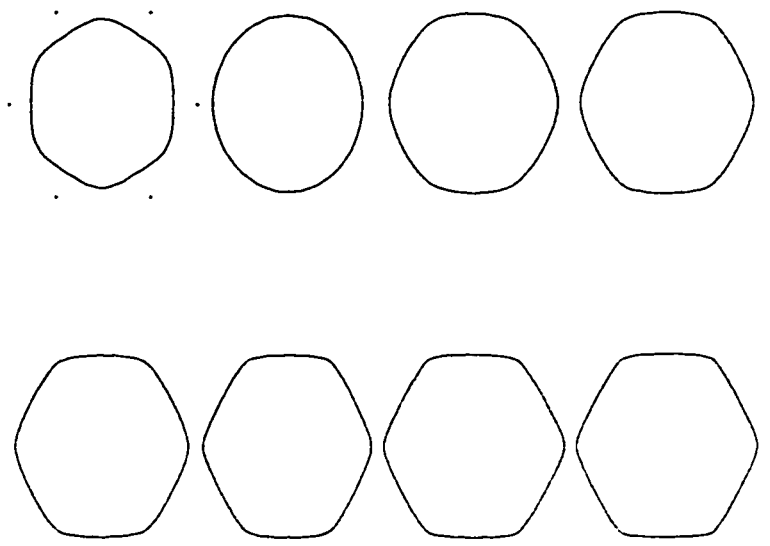

Figure 2.7: Freeform rational cubic spline curves drawn for shape parameter range 2 to 9 (Local/Global tension, left to right).

2.3.5 Implementation Details

A B-spline formulation with cubic by quadratic basis function is implemented to generate various curve and surface shapes. The method used here produces a C^2 continuous freeform spline curve and surfaces with interval tension control (both locally and globally). This is due to the presence of shape parameters in its description.

The program is written in high level language 'c' using functions putpixel and line from Graphics library, 'Graphics.h' to draw curves and control graph. Initially x, y co-ordinates of control points, knot values and shape parameter $r_i > 2$ are read. The above equation is written in the cartesian form as follows

$$p_x(t) = R_0(\theta; r_i)F_{ix} + R_1(\theta; r_i)Q_{ix} + R_2(\theta; r_i)R_{ix} + R_3(\theta; r_i)F_{i+1,x} \tag{2.17}$$
$$p_y(t) = R_0(\theta; r_i)F_{iy} + R_1(\theta; r_i)Q_{iy} + R_2(\theta; r_i)R_{iy} + R_3(\theta; r_i)F_{i+1,y}$$

where

$$F_{ix} = \lambda_i x_{i-1} + (1 - \lambda_i - \mu_i)x_i + \mu_i x_{i+1}, \tag{2.18}$$
$$F_{iy} = \lambda_i y_{i-1} + (1 - \lambda_i - \mu_i)y_i + \mu_i y_{i+1},$$
$$Q_{ix} = (1 - \alpha_i)x_i + \alpha_i x_{i+1}, \quad Q_{iy} = (1 - \alpha_i)y_i + \alpha_i y_{i+1},$$
$$R_{ix} = \beta_i x_i + (1 - \beta_i)x_{i+1}, \quad R_{iy} = \beta_i y_i + (1 - \beta_i)y_{i+1}$$

Using the above equations co-ordinate positions on the curve are calculated and plotted on the screen using putpixel routines. The input data (control points and shape parameters) are read from an input file. Similarly the output co-ordinate

positions are written onto another file and simultaneously picture is drawn on the screen.

Effects of the different shape parameter *(r)* are analyzed and are demonstrated pictorially. The effects of increase tensions, both locally and globally are observed (see figure 2.6 and figure 2.7). The results confirm the analysis of the propositions. The use of variable tensions allows better shape control. At $r_i = 2$, the cubic curve is looser, for $r_i = 3$, its a freeform/approximate spline curve.

2.3.6 Concluding Remarks

- As the increment value of knot (t) between intervals (t_i, t_{i+1}) decreases, the smoothness of the curve increases. The knot values are to be given in increasing order.
- As the value of shape parameter (r_i) increases, the tension increases, at $r_i \geq$ 10.0 for i = 0,1,... n, the curve almost becomes the control polygon.
- The curve always lies within the convex hull of control points.

2.4 Conics

Quadratics (Conics) are particularly useful in specialized applications such as molecular modeling and have also been integrated into solid modeling systems.

Rational Cubic curves represent quadratic (conic) sections, rational bicubic surfaces can represent quadrics, Therefore, the implicit quadratic equation is an alternative to rational surfaces, if only quadric surfaces are being represented [5]. Other reasons for using quadratic (conic) include ease of computation, calculating intersection of one curve with another, testing whether a point is on the curve and surface.

The following section reviews the rational quadratic B-spline (conics) developed in [2]. The idea here is basically to use this method for the conversion of rational cubic B-spline developed in [7] into quadratics (Conics).

2.4.1 Conic Spline Curve

This section revisits the structure of the freeform C^1 conic spline (B-spline representation). A B-spline function with Quadratic by linear function is implemented to generate various curve shapes [2]. The method used here produces a C^1 continuous freeform spline curve with point tension control (both locally and globally) due to the presence of shape parameters in its description.

2.4.2 Local Support Rational Quadratic Spline

According to Sarfraz [2], "For the purpose of analysis, additional knots are introduced in the interval $[t_0, t_n]$ defined by $t_{-1} < t_0$ and $t_n < t_{n+1} < t_{n+2} < t_{n+3}$.

Let α_i, $\beta_i > 0$, i=0,...,n+2 be shape parameters defined on the extended partition $t_{-1} <, \ldots, < t_{n+2} < t_{n+3}$. Let $h_i = t_{i+1} - t_i$, and $\theta \equiv \theta(t) = \frac{(t_i - t_{i+1})}{h_i}$, α_i, β_i are shape parameters. A rational quadratic (conic) spline function $\phi_j(t)$, j = 0,...,n+2 (see

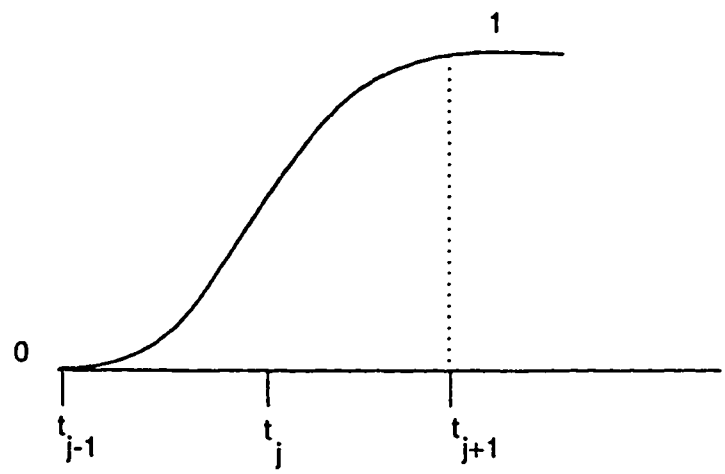

Figure 2.8: Basis function ϕ_j(t).

figure 2.8) is constructed such that

$$\phi_j(t) = \begin{cases} 0, for \ \ t < t_{j-1} \\ 1, for \ \ t \geq t_{j+1} \end{cases}$$

where t_j are the knot values. On the two intervals $[t_i, t_{i+1}]$, i=j-1,j. $\phi_j(t)$ will have the Q/L-form

$$\phi_j(t) = R_0(\theta, \alpha_i, \beta_i)\phi_j(t_i) + R_1(\theta, \alpha_i, \beta_i)V_{j,i} + R_2(\theta, \alpha_i, \beta_i)\phi_{j+1}(t_i), \qquad (2.19)$$

The functions $R_k(\theta; \alpha_i, \beta_i)$, k = 0,1,2 are the Bernstein Bezier weight functions.

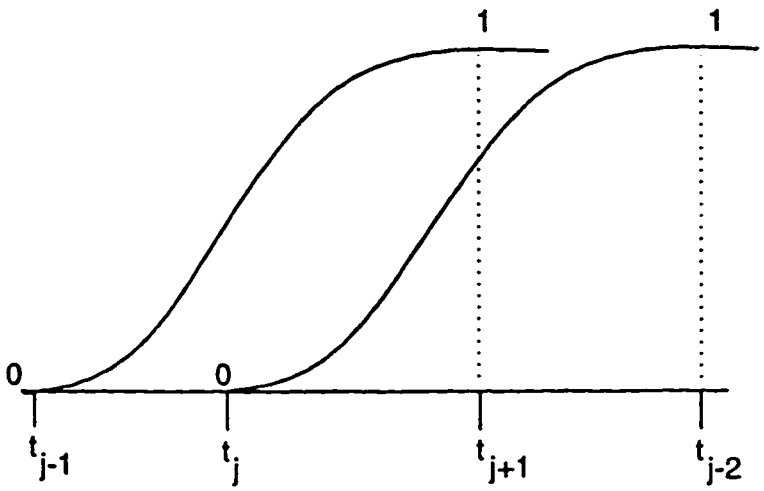

Figure 2.9: Function $\phi_j(t)$ and $\phi_{j+1}(t)$.

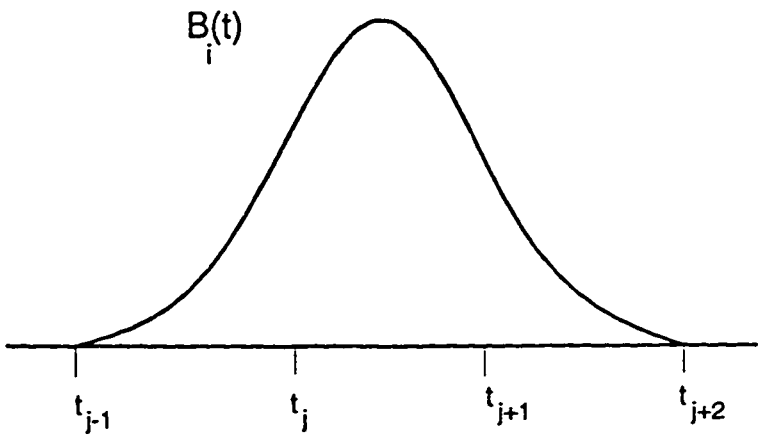

Figure 2.10: Quadratic B-spline $B_j(t)$.

Where $R_k(\theta,\alpha_i,\beta_i)$ k = 0,1,2 are defined as follows:

$$R_0(\theta,\alpha_i,\beta_i) = \frac{\alpha_i(1-\theta)^2}{Q_0(\theta,\alpha_i,\beta_i)}, \tag{2.20}$$
$$R_1(\theta,\alpha_i,\beta_i) = \frac{(\alpha_i+\beta_i)\theta(1-\theta)}{Q_0(\theta,\alpha_i,\beta_i)},$$
$$R_2(\theta,\alpha_i,\beta_i) = \frac{\beta_i\theta^2}{Q_0(\theta,\alpha_i,\beta_i)}$$

where

$$Q_0(\theta,\alpha_i,\beta_i) = \alpha_i(1-\theta)+\beta_i\theta,$$

since

$$\alpha_i(1-\theta)+\beta_i\theta = \alpha_i(1-\theta)^2+(\alpha_i+\beta_i)\theta(1-\theta)+\beta_i\theta^2$$

The requirement that $\phi_j(t)\epsilon C^1(-\infty,+\infty)$, at t_{j-1}, t_j and t_{j+1}, uniquely determines the following:

$$\hat{V}_{j,i} = \begin{cases} 1, \textit{ for } i=j, \\ 0, \textit{ otherwise} \end{cases} \tag{2.21}$$

and

$$\phi_j(t) = \mu_j \tag{2.22}$$

where

$$\mu_j = \frac{h_{j-1}\beta_{j-1}(\alpha_j + \beta_j)}{h_{j-1}\beta_{j-1}(\alpha_j + \beta_j) + h_j\alpha_j(\alpha_{j-1} + \beta_{j-1})} \tag{2.23}$$

The function $\phi_{j+1}(t)$ is constructed as shown in the figure 2.9. The local support quadratic (conic) spline function is defined by the difference functions: (see figure 2.10)

$$B_j(t) = \phi_j(t) - \phi_{j+1}(t) \tag{2.24}$$

An explicit representation of the quadratic (conic) spline $B_j(t)$ on any interval $[t_i, t_{i+1}]$ can be calculated from 2.21 and 2.22 as

$$B_j(t) = R_0(\theta; \alpha_i, \beta_i)B_j(t_i) + R_1(\theta; \alpha_i, \beta_i)V_{j,i} + R_2(\theta; \alpha_i, \beta_i)B_j(t_{i+1}). \tag{2.25}$$

where

$$B_j(t_i) = V_{j,i} = 0, \ for \ i \neq j, j+1$$

and

$$\begin{cases} B_j(t_j) = \mu_j \quad for \quad V_{j,j} = 1, \\ B_j(t_{j+1}) = 1 - \mu_j \quad for \quad V_{j,j+1} = 0 \end{cases}$$

The quadratic (conic) spline functions $B_j(t)$, j=0,...,n+1 are such that they satisfy the following properties.

- Propostion 2.1.
  - Local Support: $B_j(t) = 0$ for $t \notin (t_{j-1}, t_{j+2})$,

- Partition of Unity: $\sum_{j=-1}^{n+1} B_j(t) = 1$ for $t\epsilon[t_0, t_n]$,

The Bernstein-Bezier vertices of $B_j(t)$ are non-negative for $\alpha_i, \beta_i > 0$ and we thus have:

- Proposition 2.2.

  - Positivity $B_j(t) \geq 0$, for all t

By incorporating these blending functions, the curve representation can be expressed as

$$s(t) = \sum_{j=0}^{n+1} P_j B_j(t) \quad for \quad t\epsilon[t_0, t_n] \tag{2.26}$$

where $P_j \epsilon R^N$ define the control points of the representation and t is the parameter vector. Now, by the local support property,

$$s(t) = \sum_{j=i}^{i+2} P_j B_j(t) \quad for \quad t\epsilon[t_i, t_{i+1}] \quad i = 0, \ldots, n-1. \tag{2.27}$$

Substitution of 2.25 in the above equation then gives the piecewise defined rational Bernstein-Bezier representation.

$$s(t) = R_0(\theta, \alpha_i, \beta_i)F_i + R_1(\theta, \alpha_i, \beta_i)V_i + R_2(\theta, \alpha_i, \beta_i)F_{i+1}. \tag{2.28}$$

where

$$F_i = (1 - \mu_i)p_{i-1} + \mu_i P_i,$$

$$V_i = P_i \tag{2.29}$$

The above transformation of 2.29 can also be represented in matrix notation as

$$X_i = Y_i Z_i \tag{2.30}$$

where

$$X_i = [F_i V_i F_{i+1}]^T, \; Z_i = [P_{i-1}, P_i P_{i+1}]^T$$

and

$$Y_i = \begin{bmatrix} 1-\mu_i & \mu_i & & \\ & 1 & & \\ & & 1-\mu_{i+1} & \mu_{i+1} \end{bmatrix}$$

The transformation to rational Bernstein-Bezier form is very convenient for computational purposes".

The above function is implemented by fixing β_i=1 and changing α_i's and also by varying both shape parameters. Different curve shapes are drawn by changing location of control points. The shape properties are now examined in the following propositions as mentioned by Sarfraz [2]:

- "Proposition 2.3.
  - Point tension property: Let α_i be as assumed in the previous section and $\alpha_k \; \infty$ for some k, $1 \leq k \leq$n. Then the following holds

$$P(t_{k+1}) - P_k = \sum_{j=0}^{n+1} (P_j - P_k) B_j(t_{k+1}) \quad = (P_{k+1} - P_k) B_k(t_{k+1}) \tag{2.31}$$

by the local support property

$$= (P_{k+1} - P_k)\mu_{k+1}.$$

It can be simply shown that μ_{k+1}=0 for α_k=∞.

- Proposition 2.4
  - Interval tension property. Proposition 2.3 shows that if $\alpha_k \rightarrow \infty$, then the part of the design curve is pulled towards the control point P_k. Similarly, if $\alpha_{k+1} \rightarrow \infty$, then the design curve is pulled towards the control point P_{k+1}: this lead to the interval tension property. This could be proved directly by studying the behavior of the Bernstein Bezier control points in 2.29. This approach can be followed to prove the following:
- Corollary 1.(Global tension property)
  - Let $\alpha_i \geq \alpha > 0$, i = -1,...,n+2, and let P denote the rational B-spline control polygon, defined explicitly on $[t_i, t_{i+1}]$, i = 0,...,n, by

$$P(t) = (1 - \theta)P_i + \theta P_{i+1}, \quad \theta \equiv \theta(t) = (t - t_i)/h_i.$$

Then the Q/L-B-spline representation converges uniformly to P on $[t_0, t_{n+1}]$ as r$\rightarrow \infty$".

2.4.3 Implementation Details

The program is written in high level language 'c' using functions putpixel and line from graphics library. 'Graphics.h' to draw curve and control graph. Initially x, y

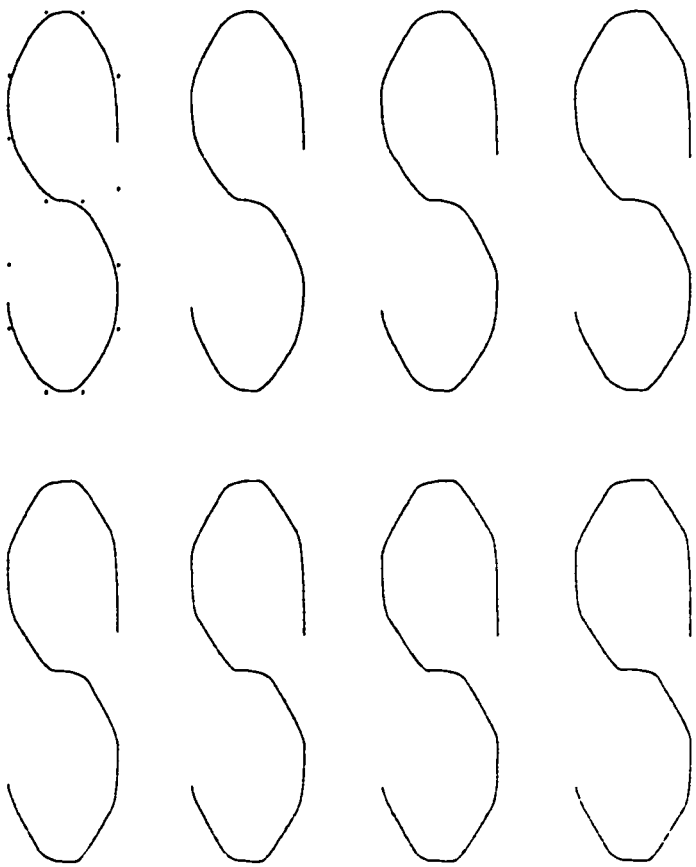

Figure 2.11: Freeform Quadratic Splines (Local/Global tension).

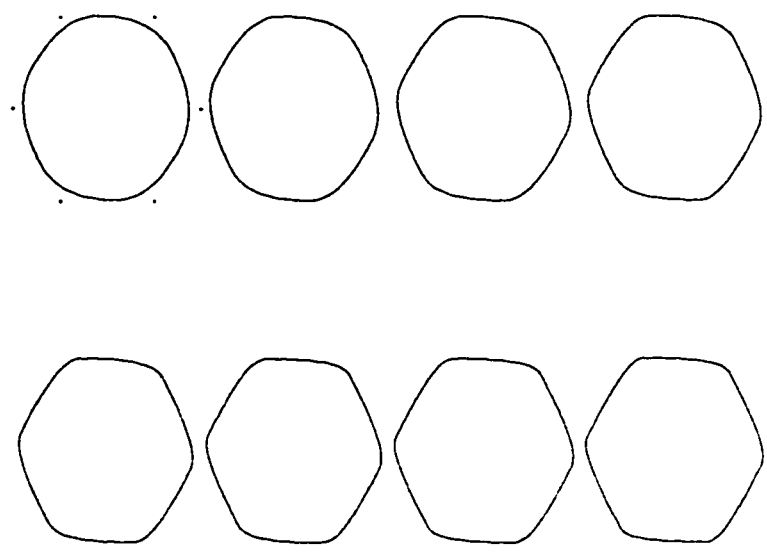

Figure 2.12: Freeform Quadratic Splines (Local/Global tension).

coordinates of control points, Knot values and shape parameters (α) are read. The equation can be written in the cartesian coordinate form as

$$x(t) = R_0(\theta, \alpha_i, \beta_i)F_{ix} + R_1(\theta, \alpha_i, \beta_i)V_{ix} + R_2(\theta, \alpha_i, \beta_i)F_{i+1,x} \tag{2.32}$$
$$y(t) = R_0(\theta, \alpha_i, \beta_i)F_{iy} + R_1(\theta, \alpha_i, \beta_i)V_{iy} + R_2(\theta, \alpha_i, \beta_i)F_{i+1,y}$$

where

$$F_{ix} = (1-\mu_i)x_{i-1} + \mu_i x_i \ ; F_{iy} = (1-\mu_i)y_{i-1} + \mu_i y_i \tag{2.33}$$
$$F_{i+1x} = (1-\mu_{i+1})x_i + \mu_{i+1}x_{i+1}, \ ; F_{i+1y} = (1-\mu_{i+1})y_i + \mu_{i+1}y_{i+1}$$
$$V_{ix} = x_i \ ; V_{iy} = y_i$$

Using the above equations co-ordinate positions on the curve are calculated and plotted on the screen using putpixel routines. Also the control points are joined by line routine to get the control graph. The input data including control points and the shape parameters are read from a input file. Similarly the output co-ordinate positions are written onto another file and simultaneously picture is also drawn on the screen.

Effects of shape parameter (α_i=2 to 9 and β_i=1), are illustrated in figure 2.11 and figure 2.12. At $\alpha_i = 2$ a freeform/approximate quadratic spline curve is obtained, as the tension (α_i) is increased, the curve approaches a control polygon shape. Effect of local/global tension are demonstrated in figure 2.12.

2.4.4 Concluding Remarks

- As the increment value of knot (t) between interval (t_i, t_{i+1}) decreases, the smoothness of the curve increases.

- As the value of shape parameter (α_i) increases the tension increases. At α_i=10.0 for i=0,1,... n, the curve becomes the control polygon.

- The curve always lies within the convex hull of control points.

- When $\alpha_{i+1} = \beta_i$, for all i, recovers the case of spline method in [23]. This case provides neither shape control on two consecutive points nor gives global tension. Thus the user cannot apply shape parameters blindly.

- If both α_i, β_i for any i, are taken equivalent, no matter as large as possible, there will not be any effect on the shape. This is because of the cancellation which converts the Q/L form into simply quadratic spline method. Thus from now onwards, one parameter is varied and other will remain fixed. Without loss of generality, $\beta_i = 1.0$, is assumed for all i, while α_i's are varied to achieve shape control.

Chapter 3

A Conic Rescue of Rational Cubic

This chapter illustrates quadratic (conic) rescue of rational cubics in [7]. It presents the advantages of quadratics (conics) achieved by rescue from rational cubic, gives detail description of rescue approach used. The chapter concludes with the conversion of rational cubics into quadratics (conics).

3.1 Overview

One of the most accepted and elegant techniques used to represent input shape information has been developed by Bezier. Pratt [24] has shown that attractive founts can also be produced using quadratic (conic) splines. Pratt showed how quadratic (conic) splines could be used to approximate some of the properties of cubic curves.

A four-point (cubic) Bezier curve consists of two control points and two endpoints. The positioning of the control points governs both the overall shape of the curve segment and the respective endpoint tangents. An inherent property of Bezier

cubic splines [24, 25], is that they allow curvature (C^2) continuity between curve segments.

Conic (ie., quadratic) curve segment can also be used to produce a required shape. The method developed by Pratt [24] forms a clean and simple way of representing quadratic (conic) sections. As any quadratic (conic) section can be achieved by a combination of a single control point, it gives added value for using quadratics (conics) rather than cubic splines. Since positioning of the control point governs the shape of a curve segment, techniques of using the shape parameters can be used to achieve it.

In the case of Bezier cubic curves, only the positioning of the control point governs the shape of a curve segment, much advanced techniques can be used to automate the use of Bezier cubic splines to yield a desired input shape. Another feature of cubic curves is that they offer eight degrees of freedom compared to seven (including the sharpness parameter offered by quadratic (conics)). This doesn't hinder the value of conics. The extra degree leads to the positioning of shapes such as cusps and loops which can also be approximated by two or more conic segments.

Conics having a solid and historical background of usage, has extra benefit of it, namely it is much easier to find an intersection of a line with a quadratic (conic) than with a cubic. The solution of it is required in many application of *Computer Graphics* e.g., hidden surface removal. Its demand also lead to the development of recursive subdivision techniques for cubic splines.

In conclusion, it can be noted that most of the advantages offered by Bezier cubic splines can equally be matched by quadratic splines. Indeed Pratt [24] has shown that the main features of cubic curves, such as

- Curvature continuity between curve segments and
- Zero curvature at inflection points

can to a good degree be approximated by quadratics (conics). Furthermore, conic splines offer a simple and feasible way of representing fount outlines as mentioned in [24] with added shape control via a sharpness parameter. To prefer conics for generating curves, the inherent properties of cubic splines have to be sought.

Before we proceed describing the rescue approach. A simple rational cubic B-spline developed by Sarfraz in [8] is discussed and converted into rational quadratic B-spline, the C^1 continuity case is presented.

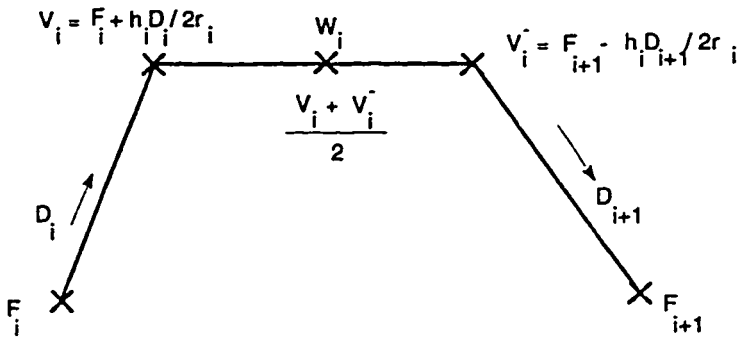

Figure 3.1: Generating Control Points.

3.2 Conversion of Rational Cubic into Conics

Let S(t) be a rational cubic spline curve of the following form (see figure 3.2):

$$S(t) = \frac{F_i(1-\theta)^3 + (F_i y_i + h_i D_i)\theta(1-\theta)^2 + (r_i F_{i+1} - h_i D_{i+1})\theta^2(1-\theta) + F_{i+1}\theta^3}{(1-\theta)^3 + r_i\theta(1-\theta)^2 + r_i\theta^2(1-\theta) + \theta^3} \tag{3.1}$$

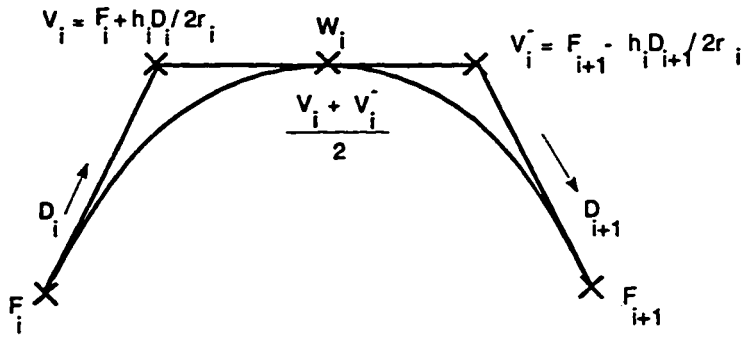

Figure 3.2: A Cubic Bezier Curve.

where $\theta \equiv \theta(t) = \frac{(t-t_i)}{h_i}$, $h_i = t_{i+1} - t_i$, and D_i's are derivatives at the points $F_i = (x_i, y_i)$ for i=1,2,...n. The spline S(t) is converted into quadratic in the following section, the C^1 spline conditions are considered for it.

3.2.1 C^1-Spline Case

Let

$$P_i(t) = \frac{F_i(1-\theta)^2 + r_i V_i \theta(1-\theta) + W_i\theta^2}{1+(r_i-2)\theta(1-\theta)} = \frac{D(\theta)}{N(\theta)},\ \theta = t\epsilon[t_i, t_i^*]$$

where

$$N(\theta) = 1+(r_i-2)\theta(1-\theta) = (1-\theta)^2 + r_i\theta(1-\theta) + \theta^2$$

$\theta \equiv \theta(t) = \frac{2\times(t-t_i)}{h_i}$ where $t\epsilon[t_i, t_i^*]$, and $t_i^* = \frac{(t_i+t_{i+1})}{2}$

For the C^1 spline case the first derivative of P_i(t) is considered. The first derivative of P_i(t) is given as:

$$P_i^{(1)}(t) = \frac{[D^{(1)}(\theta)N(\theta) - D(\theta)N^{(1)}(\theta)]}{h_i N(\theta)^2}$$

where

$$D^{(1)}(\theta) = (2W_i - r_iV_i)\theta + (r_iV_i - 2F_i)(1-\theta), \;\; N^{(1)}(\theta) = (r_i - 2)\{(1-\theta) - \theta\}$$

After simplification, we obtain the following form of $P_i^{(1)}(t)$,

$$\begin{aligned} P_i^{(1)}(t) &= 2\{\frac{r_i(V_i - F_i)(1-\theta)^3 + (2W_i + r_iV_i - r_iF_i - 2F_i)\theta(1-\theta)^2}{h_iN(\theta)^2} \\ &+\frac{(r_iW_i + 2W_i - r_iV_i - 2F_i)\theta^2(1-\theta) + r_i(W_i - V_i)\theta^3}{h_iN(\theta)^2}\} \end{aligned} \tag{3.2}$$

$P_i^{(1)}$(t) at t_i and t_i^* are:

$$P_i^{(1)}(t_i) = \frac{2r_i(V_i - F_i)}{h_i}$$
$$P_i^{(1)}(t_i^*) = \frac{2r_i(W_i - V_i)}{h_i}$$

Similarly, if

$$P_i^*(t) = \frac{W_i(1-\theta)^2 + r_iV_i^*\theta(1-\theta) + F_{i+1}\theta^2}{(1-\theta)^2 + r_i\theta(1-\theta) + \theta^2}, \;\; \theta = t\epsilon[t_i^*, t_{i+1}]$$

$\theta \equiv \theta(t) = \frac{2\times(t-t_i)}{h_i}$ where $t\epsilon[t_i^*, t_{i+1}]$, and $t_i^* = \frac{(t_i+t_{i+1})}{2}$. The first derivative of $P_i^*(t)$ after simplification obtained is of the following form:

$$\begin{aligned} P_i^{*(1)}(t) &= 2(\frac{r_i(V_i^* - W_i)(1-\theta)^3 + (2F_{i+1} + r_iV_i^* - r_iW_i - 2W_i)\theta(1-\theta)^2}{h_iN(\theta)^2} \\ &+\frac{(r_iF_{i+1} + 2F_{i+1} - r_iV_i^* - 2W_i)\theta^2(1-\theta) + r_i(F_{i+1} - V_i^*)\theta^3}{h_iN(\theta)^2}) \end{aligned} \tag{3.3}$$

and $P_i^{*(1)}(t)$ at t_i^* and t_{i+1} are

$$P_i^*(t_i^*) = \frac{2r_i(V_i^* - W_i)}{h_i},$$
$$P_i^*(t_{i+1}) = \frac{2r_i(F_{i+1} - V_i^*)}{h_i},$$

Let D_i, i=1,2,...,n be the derivatives at control points, then the requirements for the curve to be C^1 continuous are: (see figure 3.1 and figure 3.2).

$$P_i^{(1)}(t_i) = D_i \Rightarrow V_i = F_i + \frac{h_i D_i}{2r_i} \tag{3.4}$$

$$P_i^{*(1)}(t_{i+1}) = D_{i+1} \Rightarrow V_i^* = F_{i+1} + \frac{h_i D_{i+1}}{2r_i} \tag{3.5}$$

From equation 3.4, 3.5 and the constraint

$$P_i^{(1)}(t_i^*) = P_i^{*(1)}(t_i^*)$$

follows

$$\frac{2r_i(W_i - V_i)}{h_i} = \frac{2r_i(V_i^* - W_i)}{h_i},$$

$$W_i = \frac{(V_i + V_i^*)}{2} = \frac{(F_i + F_{i+1})}{2} + \frac{h_i}{4r_i}(D_i - D_{i+1}) \tag{3.6}$$

substitution of equations 3.4, 3.5 and 3.6 in equation 3.2 and 3.3 simplifies as:

$$P_i^{(1)}(t) = \frac{D_i(1-\theta)^3 + A_{1,i}\theta^2(1-\theta) + A_{2,i}\theta(1-\theta)^2 + A_{3,i}\theta^3}{h_i N(\theta)^2} \tag{3.7}$$

and

$$P_i^{*(1)}(t) = \frac{A_{3,i}(1-\theta)^3 + A_{4,i}\theta(1-\theta)^2 + A_{5,i}\theta^2(1-\theta) + D_{i+1}\theta^3}{h_i N(\theta)^2} \tag{3.8}$$

The curve is monotonic increasing if and only if $A_{j,i} \geq 0$ for all j=1,2,...,5, $t\epsilon[t_i, t_{i+1}]$, for all i where

$$A_{1,i} = 2\delta_i + D_i - (D_{i+1} - D_i)/r_i$$

$$A_{2,i} = r_i\delta_i - \frac{D_{i+1} + D_i}{2} + 2\delta_i + D_i - \frac{D_{i+1} - D_i}{r_i} = A_{3,i} + A_{1,i}$$

$$A_{3,i} = r_i\delta_i - \frac{D_{i+1} + D_i}{2}$$

$$A_{4,i} = A_{3,i} + A_{5,i}$$

$$A_{5,i} = D_{i+1} + 2\delta_i + \frac{D_{i+1} - D_i}{r_i}$$

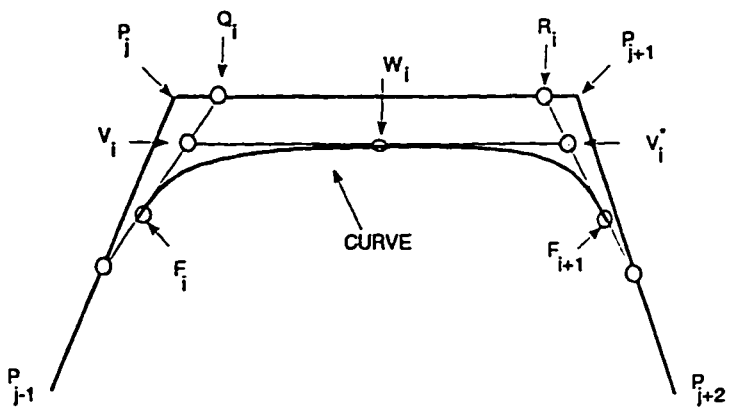

Figure 3.3: A Cubic B spline curve

Figure 3.4: Freeform Rational Cubic Curves.

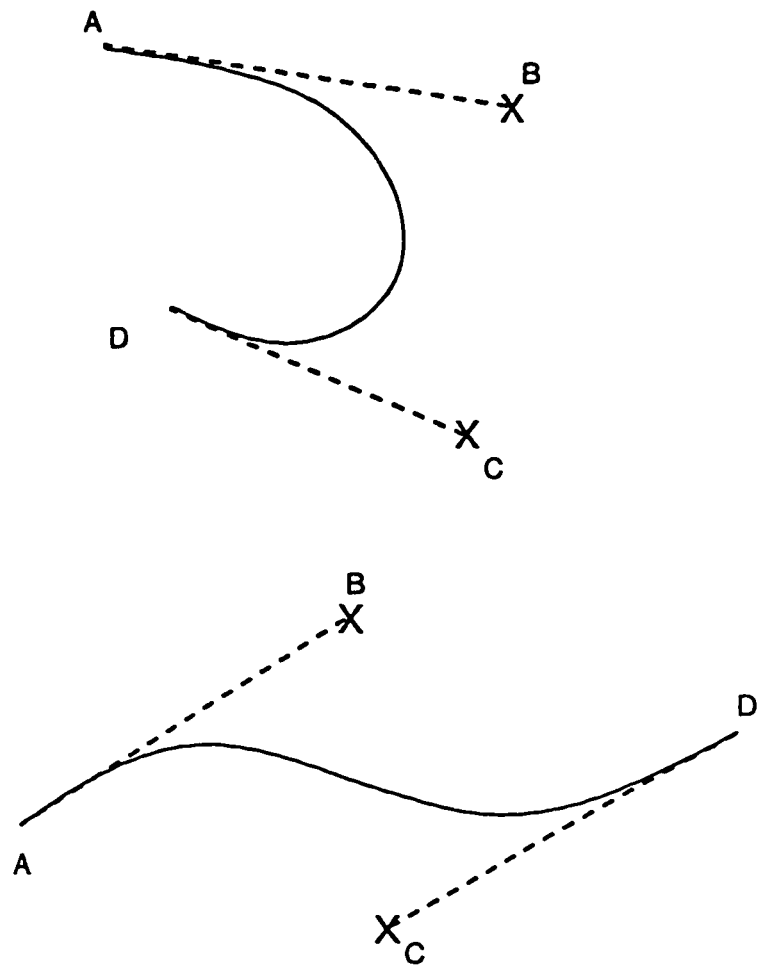

Figure 3.5: Examples of Bezier Curves, where A & D are endpoints, and B & C are control points.

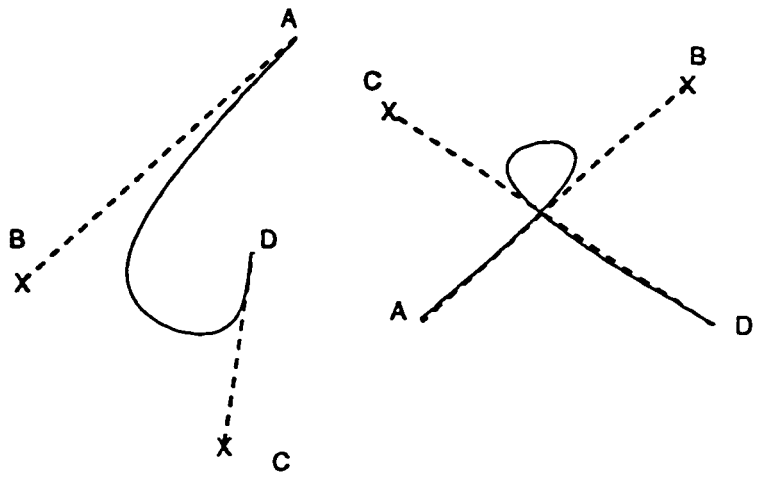

Figure 3.6: Examples of Bezier Curves, where A & D are endpoints, and B & C are control points.

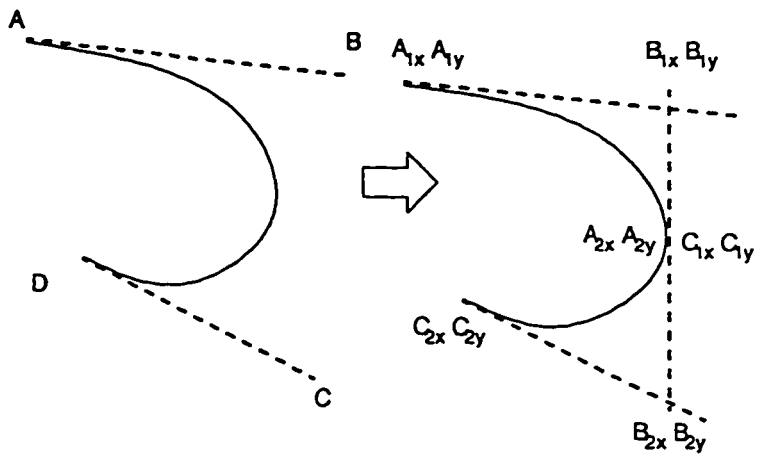

Figure 3.7: Figure shows possibility of Conic rescue of Bezier curves, when the end point tangents do not intersect, the Bezier is divided at $t = t_i^*$ to form two conics.

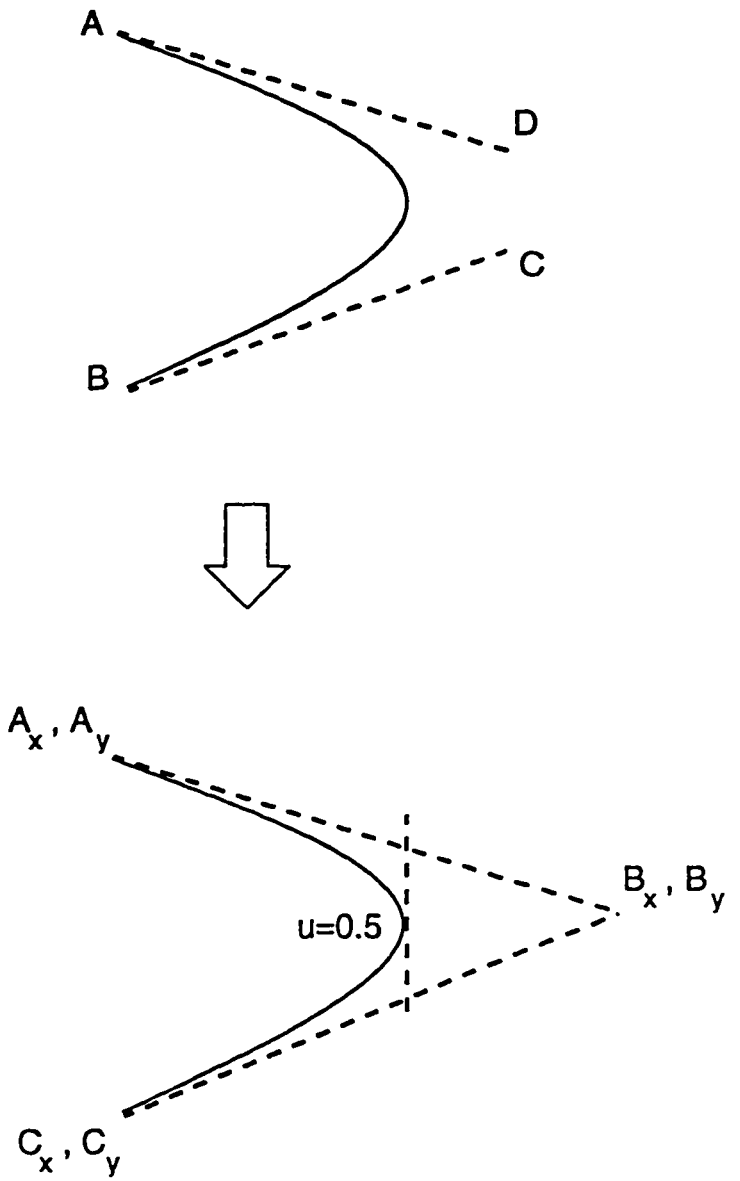

Figure 3.8: When two end point tangent do intersect.

3.3 A Freeform Rational Cubic Spline

The curve in figure 3.3 represents a B-spline freeform/approximate curve with control points $P_{j-1}, P_j, P_{j+1}, P_{j+2}$. The design curve of the B-spline is given as:

$$p(t) = \sum_{j=-1}^{n+1} P_j B_j(t), \quad t\epsilon[t_0, t_n], \tag{3.9}$$

where $P_j \epsilon R^N$ are the control points. A piecewise rational cubic Bernstein-Bezier representation of the design curve 3.9 is of the form as in 2.11. A freeform rational cubic spline curves obtained is of the form as shown in figure 3.4 for shape parameter $r_i = 3$ using the same approach as in 2.11.

3.4 A Conic Rescue of Rational Cubic

A four-point (cubic) Bezier curve consist of two control points and two endpoints. The positioning of control points governs both the overall shape of the curve segment and there tangents at the endpoints. Few examples of the Bezier cubic curves are depicted in figures 3.5 and 3.6. The above rational Cubic B-spline is transformed to the rational quadratic (conic) B-spline by identifying the points as follows:

3.4.1 Rescue Realization

The process of rescue realization is taken from [7]. The problem of fitting the conic curve to a given rational cubic curve can be considered as a general problem of fitting the best curve to a given set of data points.

When deciding to recover the rational cubic curve, described by cubic curve

segments. The following possibilites are considered (see figures 3.7 and 3.8). When the two end point tangents of the rational cubic curve segment do intersect or not. The rational cubic curve segment is divided into two halves, with two different curves representing each half. The rational cubic curve is divided at $t_i = t_i^*$, where $t_i^* = \frac{t_i+t_{i+1}}{2}$. This division doesn't guarantee that splitting at this point will always result in conic sections that best fit the original rational cubic curve.

But this choice provides reasonable recovery of the rational cubics. Some other choices may also be discussed.

The rational quadratics (conics) will be constructed with the following points obtained by drawing tangents at the end points of the rational cubic curve segment (see figure 3.3).

$$F_i = \lambda_i P_{i-1} + (1 - \lambda_i - \mu_i)P_i + \mu_i P_{i+1} \tag{3.10}$$

$$V_i = \frac{(F_i + Q_i)}{2} \tag{3.11}$$

$$V_i^* = \frac{(R_i + F_{i+1})}{2} \tag{3.12}$$

$$W_i = \frac{(V_i + V_i^*)}{2} \tag{3.13}$$

3.5 A Freeform Rational Conic Spline

The rational quadratics (Conics) constructed are of the following form:

$$S(t) = \begin{cases} R_0(\theta; r_i)F_i + R_1(\theta; r_i)V_i + R_2(\theta; r_i)W_i, & for \ \ \theta = t \epsilon [t_i, \frac{t_i+t_{i+1}}{2}] \\ R_3(\hat{\theta}; r_i)W_i + R_4(\hat{\theta}; r_i)V_i^* + R_5(\hat{\theta}; r_i)F_{i+1}, & for \ \ \hat{\theta} = t \epsilon [\frac{t_i+t_{i+1}}{2}, t_{i+1}] \end{cases}$$

where $R_k(\theta; r_i)$ k=0,1,2 are defined as

$$R_0(\theta; r_i) = \frac{(1-\theta)^2}{D_0(\theta, r_i)} \tag{3.14}$$

$$R_1(\theta; r_i) = \frac{r_i\theta(1-\theta)}{D_0(\theta, r_i)} \tag{3.15}$$

$$R_2(\theta; r_i) = \frac{\theta^2}{D_0(\theta, r_i)} \tag{3.16}$$

where

$$D_0(\theta; r_i) = 1 + (r_i - 2)\theta(1-\theta) \tag{3.17}$$

since

$$1 + (r_i - 2)\theta(1-\theta) = (1-\theta)^2 + r_i\theta(1-\theta) + \theta^2 \tag{3.18}$$

The functions $R_k(\theta; r_i)$, k=0,1,2 are actually the Bernstein-Bezier weight functions. Similarly $R_k(\hat{\theta}; r_i)$ k=3,4,5 are defined as

$$R_3(\hat{\theta}; r_i) = \frac{(1-\hat{\theta})^2}{D_0(\hat{\theta}, r_i)} \tag{3.19}$$

$$R_4(\hat{\theta}; r_i) = \frac{r_i\hat{\theta}(1-\hat{\theta})}{D_0(\hat{\theta}, r_i)} \tag{3.20}$$

$$R_5(\hat{\theta}; r_i) = \frac{\hat{\theta}^2}{D_0(\hat{\theta}, r_i)} \tag{3.21}$$

where

$$D_0(\hat{\theta}; r_i) = 1 + (r_i - 2)\hat{\theta}(1-\hat{\theta}) \tag{3.22}$$

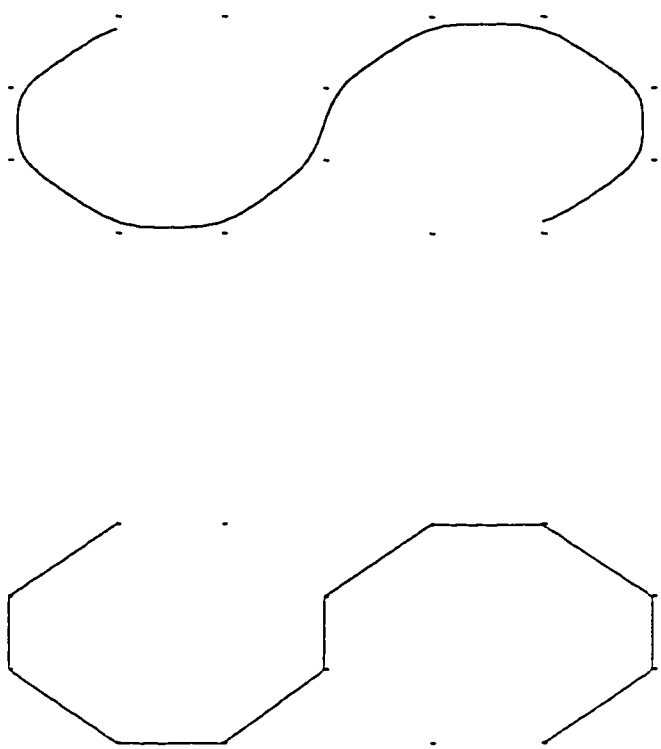

Figure 3.9: A Freeform Rational Conic Curve drawn as control polygon and as single curve ($r_i = 2$).

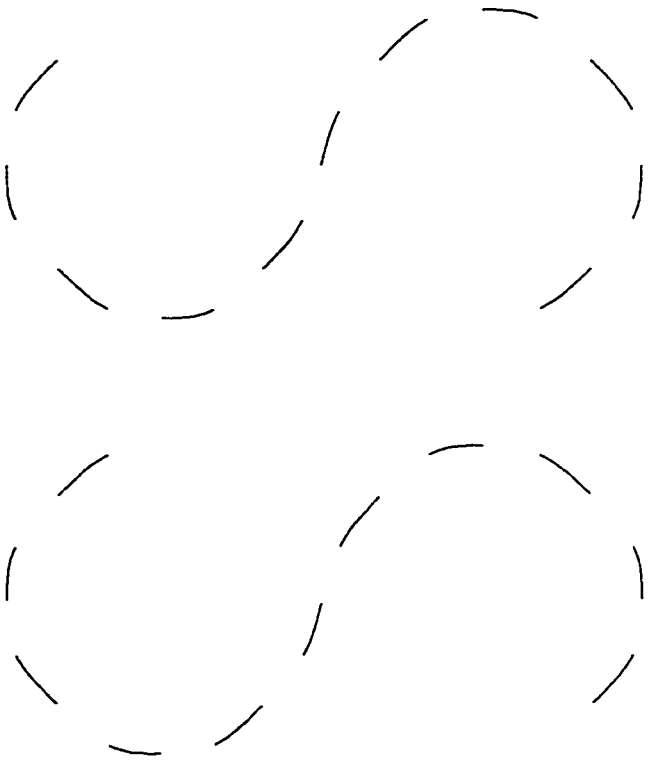

Figure 3.10: A Freeform Rational Conic Curve plotted piecewise (left half curve corresponds to first quadratic and right half to the second quadratic).

since

$$1+(r_i-2)\hat{\theta}(1-\hat{\theta}) = (1-\hat{\theta})^2 + r_i\hat{\theta}(1-\hat{\theta}) + \hat{\theta}^2 \tag{3.23}$$

The functions $R_k(\hat{\theta}; r_i)$, k=3,4,5 are actually the Bernstein-Bezier weight functions. The spline curve is plotted as a single curve as shown in figure 3.9 for shape parameter r_i=2 and also as a control polygon. Figure 3.10 illustrates the curves plotted using the above rational quadratic (conic) spline method. The left curve corresponds to the first quadratic (conic) in the interval t_i to $\frac{t_i+t_{i+1}}{2}$, and the right curve with the second quadratic (conic) in the interval $\frac{t_i+t_{i+1}}{2}$ to t_{i+1}.

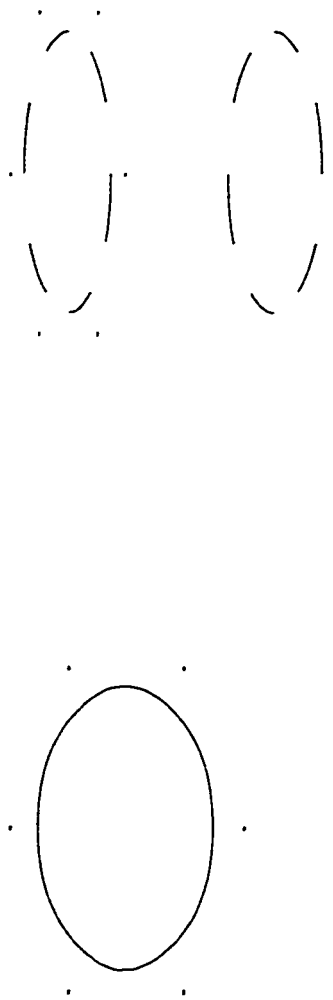

Figure 3.11: A Freeform Rational Conic ($r_i = 2$, both the left and right curves plotted as distinct and single curve).

Figure 3.12: A Freeform Rational Conic (r_i=2).

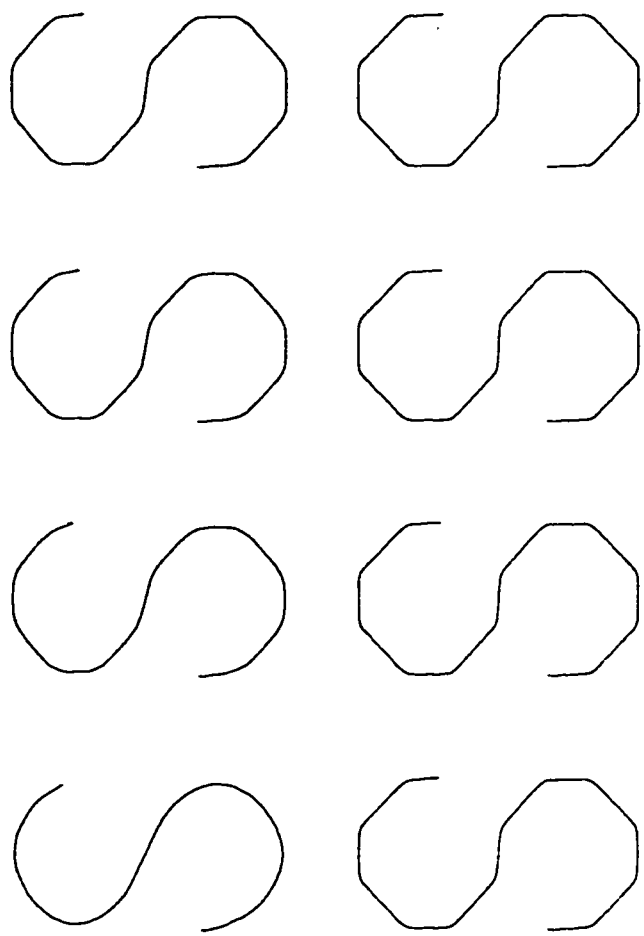

Figure 3.13: A Freeform Rational Conic Curves drawn for different shape parameters (2 to 9).

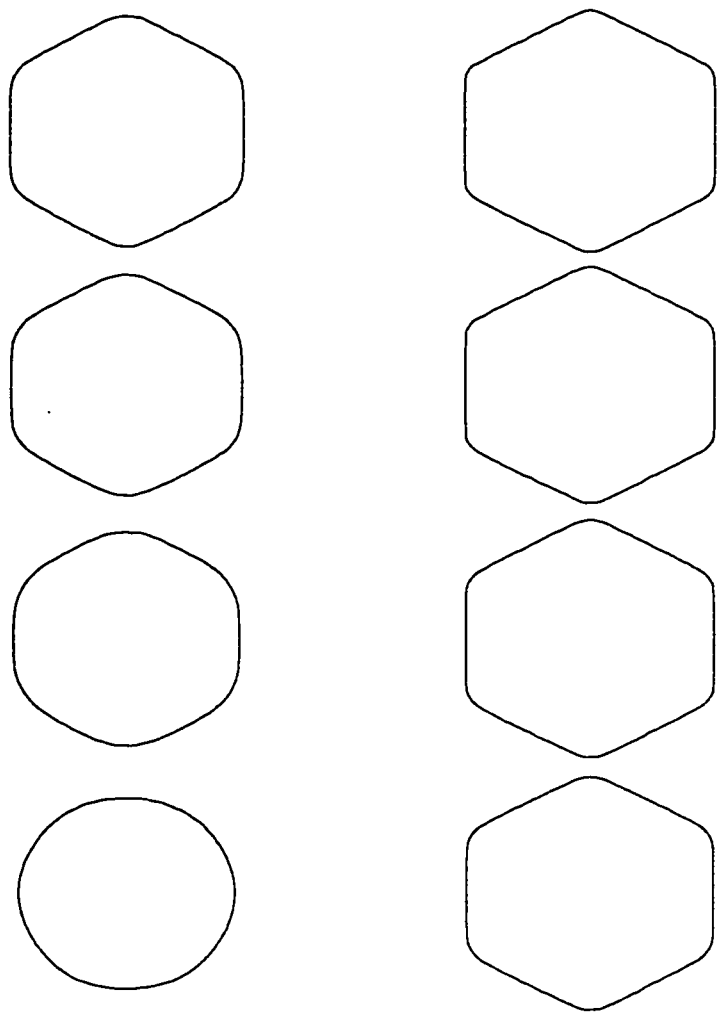

Figure 3.14: A Freeform Rational Conic Curves drawn for different shape parameters (2 to 9).

Different curves are plotted to illustrate the behaviour of quadratic (conic) spline developed. Figures 3.11 and 3.12 show the different quadratic (conic) spline curves drawn using the above method. Figure 3.13 and 3.14 shows the curves drawn with different shape parameters.

A matrix representation of the quadratics (conics) is of the following forms:

$$X_i = Y_i Z_i$$

$$\hat{X}_i = \hat{Y}_i Z_i$$

The quadratic (conic) matrix 3.24 is represented in the following form:

$$X_i = [F_i\, V_i\, W_i]^T,\ \hat{X}_i = [W_i\, V_i^*\, F_{i+1}]^T,\ Z_i = [P_{i-1}\, P_i\, P_{i+1}\, P_{i+2}]^T$$

and

$$Y_i = \begin{bmatrix} \lambda_i & 1-\lambda_i-\mu_i & \mu_i & \\ \frac{\lambda_i}{2} & \frac{(1-\lambda_i-\mu_i)+(1-\alpha_i)}{2} & \frac{\mu_i+\alpha_i}{2} & \\ \frac{\lambda_i}{4} & \frac{(1-\lambda_i-\mu_i)+(1-\alpha_i)+\beta_i+\lambda_{i+1}}{4} & \frac{(1-\lambda_{i+1}-\mu_{i+1})+(1-\beta_i)+(\mu_i+\alpha_i)}{4} & \frac{\mu_{i+1}}{4} \end{bmatrix} \tag{3.24}$$

$$\hat{Y}_i = \begin{bmatrix} \frac{\lambda_i}{4} & \frac{(1-\lambda_i-\mu_i)+(1-\alpha_i)+\beta_i+\lambda_{i+1}}{4} & \frac{(1-\lambda_{i+1}-\mu_{i+1})+(1-\beta_i)+(\mu_i+\alpha_i)}{4} & \frac{\mu_{i+1}}{4} \\ & \frac{\beta_i+\lambda_{i+1}}{2} & \frac{(1-\lambda_{i+1}-\mu_{i+1})+(1-\beta_i)}{2} & \frac{\mu_{i+1}}{2} \\ & \lambda_{i+1} & 1-\lambda_{i+1}-\mu_{i+1} & \mu_{i+1} \end{bmatrix} \tag{3.25}$$

3.6 Rational B-Spline Surfaces

In the following section we generalize the idea of the previous section to rational bicubic as in [7] and biquadratic surfaces.

3.6.1 Rational Cubic B-Splines And The Design Surface

Suppose we are given points

$$P_{i,j}\epsilon R^3, i = -1, ..., m+1, j = -1, ..., n+1, \tag{3.26}$$

and the knot sequences

$$\tilde{t}_0 < \tilde{t}_1 < ..... < \tilde{t}_m . t_0 < t_1 < ..... < t_n. \tag{3.27}$$

with appropriate additional knots

$$\begin{aligned} &\tilde{t}_{-3} < \tilde{t}_{-2} < \tilde{t}_{-1} < \tilde{t}_0;\ \ \tilde{t}_m < \tilde{t}_{m+1} < \tilde{t}_{m+2} < \tilde{t}_{m+3}; \\ &t_{-3} < t_{-2} < t_{-1} < t_0;\ \ t_n < t_{n+1} < t_{n+2} < t_{n+3}. \end{aligned} \tag{3.28}$$

A parametric rational B-spline surface $p(\tilde{t}, t)$ is find in such a way that $p(\tilde{t}, t_j)$ and $p(\tilde{t}_i, t)$ are freeform rational cubic splines with tension in $\tilde{t}$ and t-directions for all i and j respectively.

Suppose we are given tension parameters

$$\tilde{r}_{i,j} > 2\ \&\ r_{i,j} > 2, i = -3, ..., m+2, j = -3, ..., n+2. \tag{3.29}$$

with $\tilde{B}_k(\tilde{t},t)$ and $B_l(t,\tilde{t})$ the corresponding rational B-spline basis functions, as mentioned in 2.11, but with variable cubic B-spline tensions $\tilde{r}_i(t)$ and $r_j(\tilde{t})$ defined as:

$$\tilde{r}_i(t) = \sum_j \tilde{r}_{i,j} N_j(t),$$
$$r_j(t) = \sum_i r_{i,j} N_i(\tilde{t}). \tag{3.30}$$

where $N_j(t)$ are cubic B-splines. These can be computed as a special case of the rational cubic splines of 2.11.

The convex hull property of N_j show that $\tilde{r}_i(t)$, $r_j(\tilde{t}) > 2$, for all i, j, $\tilde{t}$ and t. Also, for any j

$$\tilde{r}_i(t) = 0, t \notin (t_{j-2}, t_{j+2})$$

and for any i

$$r_j(\tilde{t}) = 0, \tilde{t} \notin (\tilde{t}_{i-2}, \tilde{t}_{i+2})$$

As mentioned by Sarfraz in [7]" A large value of the shape parameter $\tilde{r}_{i,j}$ for any j results in large values of $\tilde{r}_i(t)$ in the intervals $[t_{j-1}, t_j]$, $[t_j, t_{j+1}]$ and $[t_{j+1}, t_{j+2}]$ and a significant rise is achieved in the interval $[t_j, t_{j+1}]$. A similar characteristic is possesed by $r_j(\tilde{t})$. Thus a sufficiently large value of any of the shape parameters, results in a sufficiently large value of the variable weight in the corresponding interval.

In particular, if any of $\tilde{r}_i(t_j)$ and $r_j(\tilde{t}_i)$, i = 0,....,m-1; j = 0,....,n-1 tends to infinity, it causes the corresponding values from amongst $\tilde{r}_i(t_j)$ and $r_j(\tilde{t}_i)$, i = 0,....,m-1; j = 0,....,n-1, respectively to approach infinity. Hence the shape parameters are chosen in such a way that one shape parameter is associated with each interval. These shape parameters control the surface locally and globally".

Define the surface:

$$p(\tilde{t},t) = \sum_{k=-1}^{m+1} \sum_{l=-1}^{n+1} P_{k,l} \tilde{B}_k(\tilde{t},t) \tilde{B}_l(t,\tilde{t})$$

by the local support property.

$$p(\tilde{t},t) = \sum_{k=i-1}^{i+2} \sum_{l=j-1}^{j+2} P_{k,l} \tilde{B}_k(\tilde{t},t) \tilde{B}_l(t,\tilde{t})$$

Where $\tilde{t}_i \leq \tilde{t} < \tilde{t}_{i+1}$, $t_j \leq t < t_{j+1}$, i=0,...,m-1, j=0,...,n-1. Substitution of the Bernstein-Bezier form of the rational B-splines gives the piecewise defined rational Bernstein-Bezier representation.

$$p(\tilde{t},t) = \sum_{k=0}^{3} \sum_{l=0}^{3} X_{k,l}^{i,j}(\tilde{t},t) R_k(\tilde{\theta};\tilde{r}_i(t)) R_l(\theta;\tilde{r}_j((t)), \tag{3.31}$$

The Bernstein-Bezier points $X_{k,l}^{i,j}(\tilde{t},t)$ can be computed from the rational B-spline vertices $P_{i,j}$ as

$$X_{i,j} = \tilde{Y}_i Z_{i,j} Y_j^T, \tag{3.32}$$

where,

$$X_{i,j} = \begin{bmatrix} X_{0,0}^{i,j} & X_{0,1}^{i,j} & X_{0,2}^{i,j} & X_{0,3}^{i,j} \\ X_{1,0}^{i,j} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ X_{3,0}^{i,j} & \cdots & \cdots & X_{3,3}^{i,j} \end{bmatrix} \tag{3.33}$$

and

$$Z_{i,j} = \begin{bmatrix} P_{i-1,j-1} & P_{i-1,j} & \cdots & P_{i,j+2} \\ P_{i,j-1} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ P_{i+2,j} & \cdots & \cdots & P_{i+2,j+2} \end{bmatrix} \tag{3.34}$$

and the matrix Y_j is given as in 3.24 as well as $\tilde{Y}_i$ provided tildes are put where appropriate. $\tilde{Y}_i$ and Y_j^T now depend on $\tilde{t}$ and t respectively. Let

$$\begin{aligned} p(\tilde{t}, t_j) &= f_j(\tilde{t}) \\ p(\tilde{t}_i, t) &= \tilde{f}_i(t) \end{aligned} \tag{3.35}$$

Then it can be noted that each of 3.35 is a freeform rational spline with tension parameters $\tilde{r}_i(t_j)$ and $\tilde{r}_j(t_i)$ in each of the intervals $[\tilde{t}_i, \tilde{t}_{i+1}]$ and $[t_j, t_{j+1}]$ respectively, moreover.

$$f_j(\tilde{t}_i) = X_{k,l}^{i,j}(\tilde{t}_i, t_j) = \tilde{f}_i(t_j) \tag{3.36}$$

Thus the surface is constructed in such a way that it allows different interval tensions.

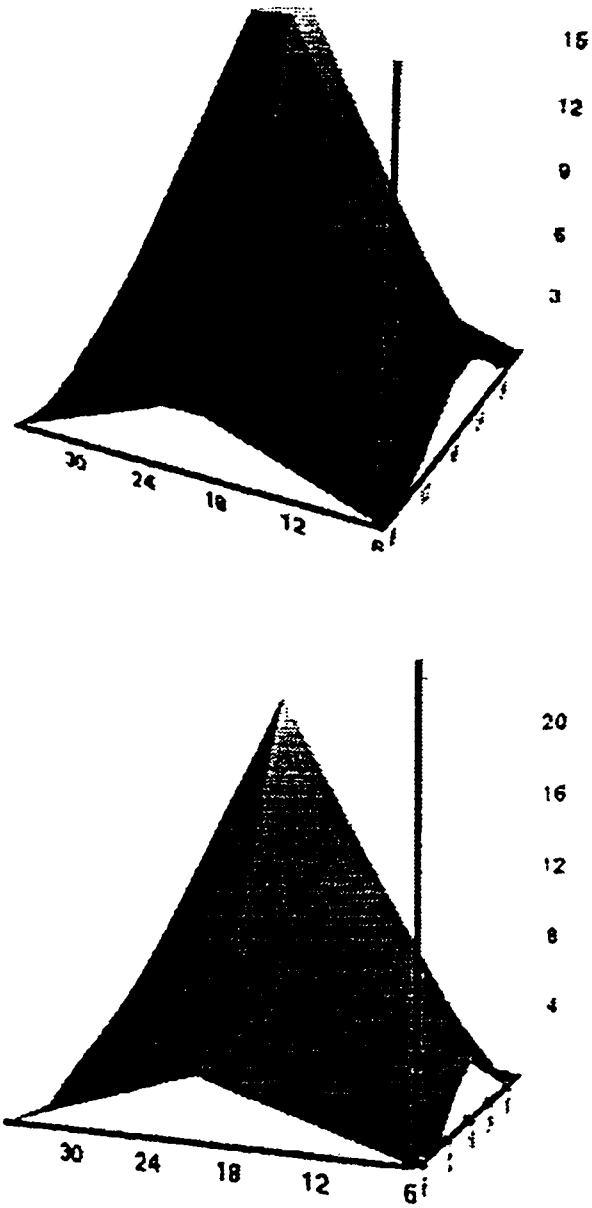

Figure 3.15: A Freeform Rational Bicubic Surface and the effect of Global tension (r_i=3 & 50).

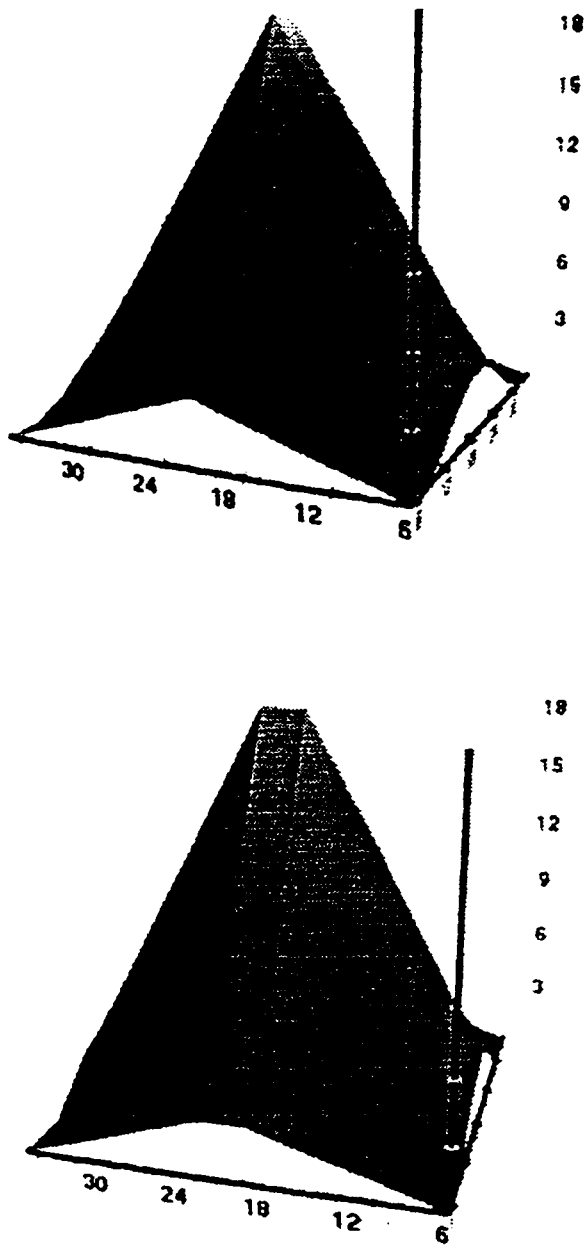

Figure 3.16: Tension along $\tilde{t}$ and t direction (r_i=50).

3.6.2 Remarks

The above surface is a simple bivariate B-spline product method which results in a C^2 freeform surface as well as provides shape control. This method is similar to a tensor product method but actually it is not. C^2 variable tensions are introduced in the B-splines which distinguish this surface method from the tensor product method and help in producing a C^2 surface which has tension control both locally and globally (see figure 3.15). The effect of interval tension along t and $\tilde{t}$ direction is demonstrated in figure 3.16. For detail proofs of the spline properties refer [7].

3.6.3 Rational Biquadratic B-Splines And The Design Surface

Finally, a piecewise defined rational Bernstein-Bezier representation is given as:

$$p(\tilde{t},t) = \begin{cases} \sum_{k=0}^{2}\sum_{l=0}^{2} X_{k,l}^{i,j}(\tilde{t},t)R_k(\tilde{\theta};\tilde{r}_i(t))R_l(\theta;r_j(\tilde{t})) & t,\tilde{t}\epsilon[t_i,t_i^*] \\ \sum_{k=0}^{2}\sum_{l=0}^{2} \hat{X}_{k,l}^{i,j}(\tilde{t},t)R_k(\tilde{\theta};\tilde{r}_i(t))R_l(\hat{\theta};r_j(\tilde{t})), & t,\tilde{t}\epsilon[t_i^*,t_{i+1}] \end{cases} \tag{3.37}$$

where $t_i^* = \frac{t_i+t_{i+1}}{2}$. The Bernstein-Bezier points $X_{k,l}^{i,j}$, can be computed from the rational B-spline vertices $P_{i,j}$ as:

$$X_{i,j} = \begin{cases} \tilde{Y}_i Z_{i,j} Y_j^T, \\ \hat{\tilde{Y}}_i Z_{i,j} \hat{Y}_j^T \end{cases} \tag{3.38}$$

where,

$$X_{i,j} = \begin{bmatrix} X_{0,0}^{i,j} & X_{0,1}^{i,j} & X_{0,2}^{i,j} & X_{0,3}^{i,j} \\ X_{1,0}^{i,j} & \dots & \dots & \dots \\ X_{2,0}^{i,j} & \dots & \dots & X_{2,3}^{i,j} \end{bmatrix} \tag{3.39}$$

and

$$Z_{i,j} = \begin{bmatrix} P_{i-1,j-1} & \dots & \dots & P_{i,j+2} \\ P_{i,j-1} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ P_{i+2,j} & \dots & \dots & P_{i+2,j+2} \end{bmatrix} \tag{3.40}$$

and the matrix Y_i and $\hat{Y}_i$ are as given in 3.24, 3.25 as well as $\tilde{Y}_i$, provided tildes are placed where appropriate. $\tilde{Y}_i$ and Y_j^T now depend on $\tilde{t}$ and t respectively. Let

$$\begin{aligned} p(\tilde{t}, t_j) &= f_j(\tilde{t}) \\ p(\tilde{t}_i, t) &= \tilde{f}_i(t) \end{aligned} \tag{3.41}$$

then it can be noted that each of 3.41 is a freeform rational spline with tension parmeters $\tilde{r}_i(t_j)$ and $r_j(\tilde{t}_i)$ in each of the intervals $[\tilde{t}_i, \tilde{t}_{i+1}]$, $[t_j, t_{j+1}]$ respectively. Thus, the surface is constructed in such a way that it allows different interval tensions.

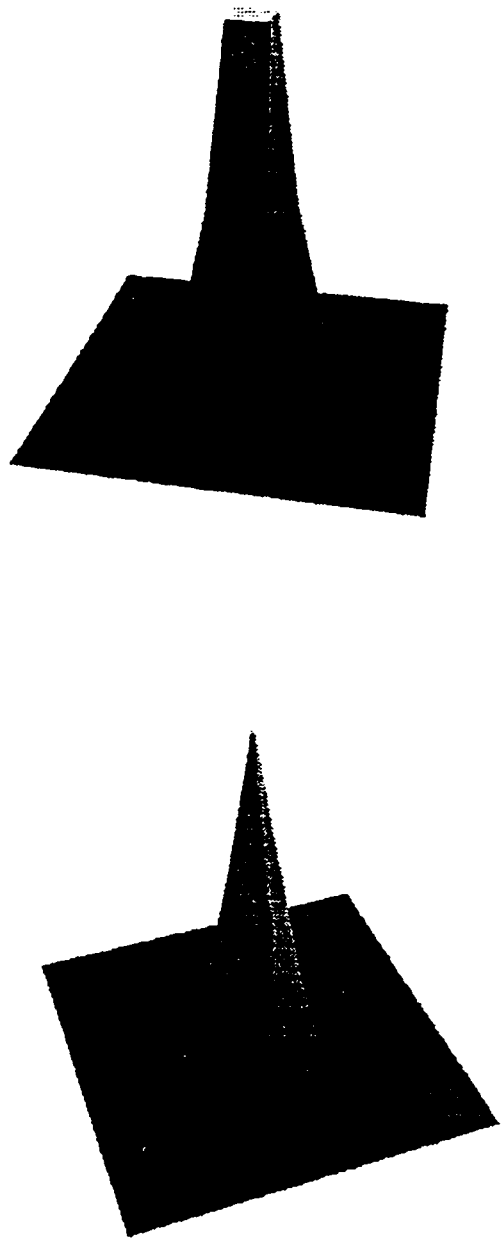

Figure 3.17: A Biquadratic Surface (Freeform and the effect of Global tension (r_i=3 & 50)).

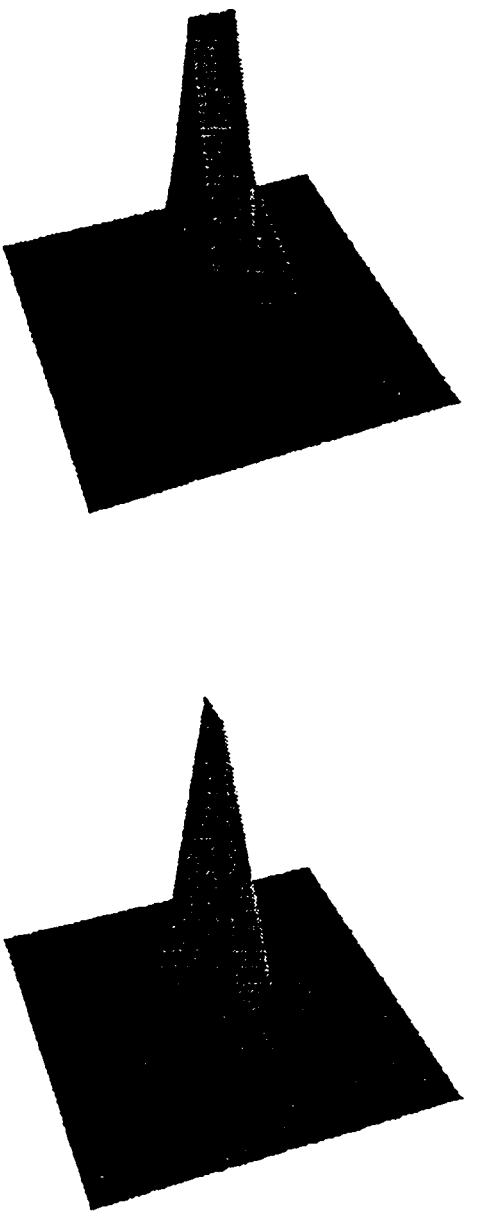

Figure 3.18: Tension along $\tilde{t}$ and t direction (r_i=50).

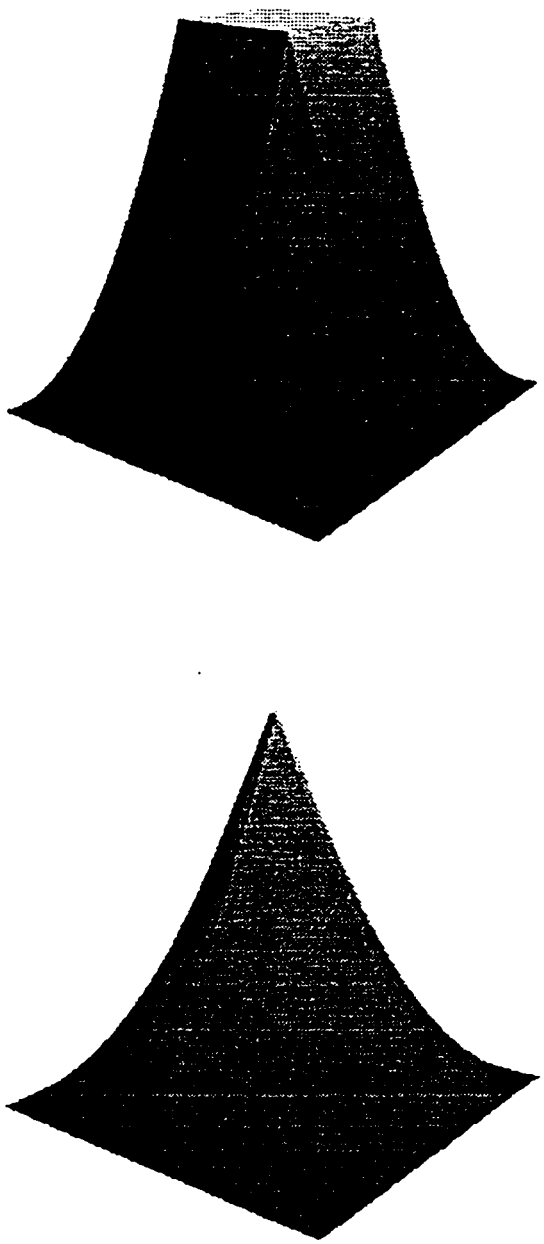

Figure 3.19: A Freeform Biquadratic Surface and the effect of Global Tension (r_i=3 & 50).

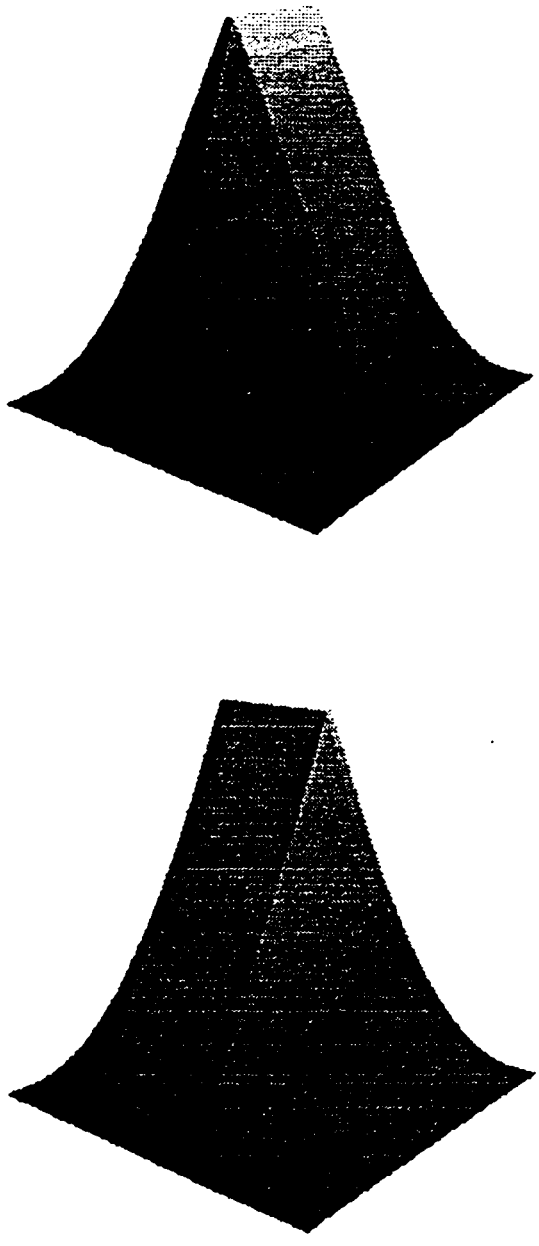

Figure 3.20: Effect of tension along $\tilde{t}$ and t direction (r_i=50).

3.6.4 Remarks

The figures 3.17 and 3.19 demonstrates the effect of global and local tension, the figures 3.18 and 3.20 shows the effect of increase in tension along t and $\tilde{t}$ direction.

Chapter 4

Comparison, Conclusion and Future Work

4.1 Overview

This chapter compares the rational Cubic B-splines [7] with rational Conic B-splines obtained from conversion, presents conclusions and gives directions for future work.

4.2 Comparison of Rational Cubics and Rational Quadratics (Conics)

C^1 continuous quadratic B-splines are obtained as an alternative to the rational cubic B-splines and different curves are drawn to illustrate the comparison between them.

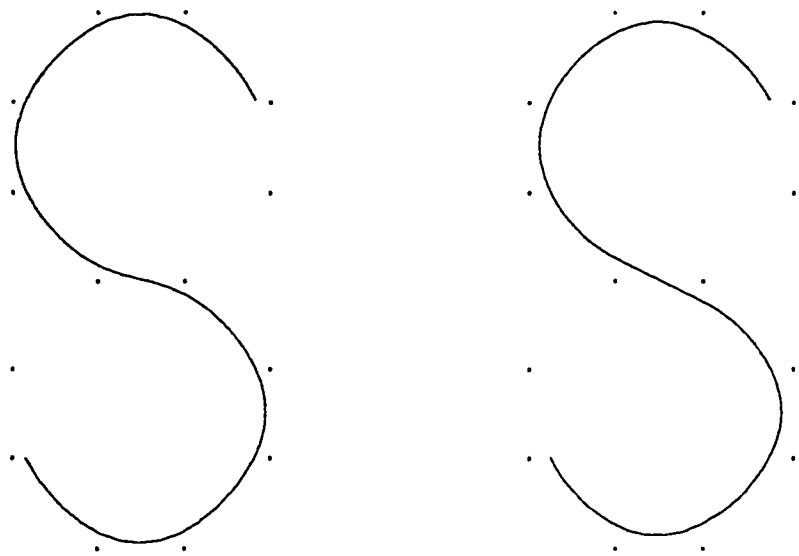

Figure 4.1: Cubic ($r_i = 3$) and Conic Spline Curves ($r_i = 2$).

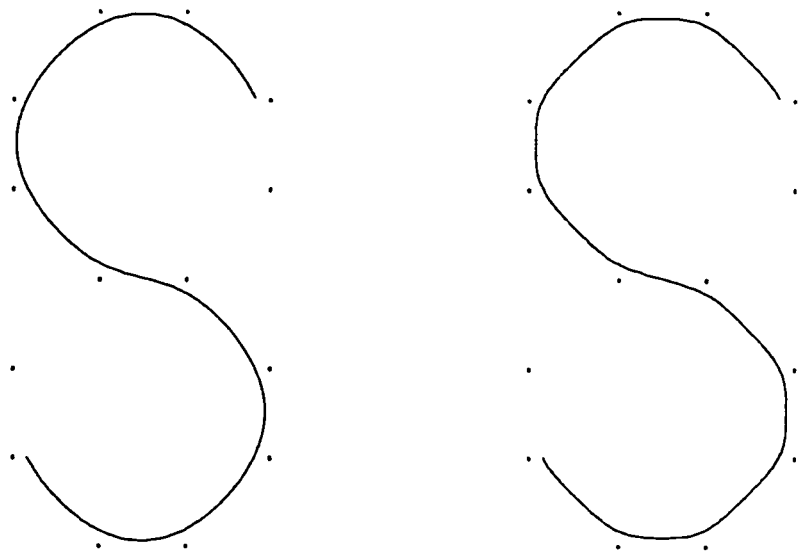

Figure 4.2: Cubic and Conic Spline Curves ($r_i = 3$).

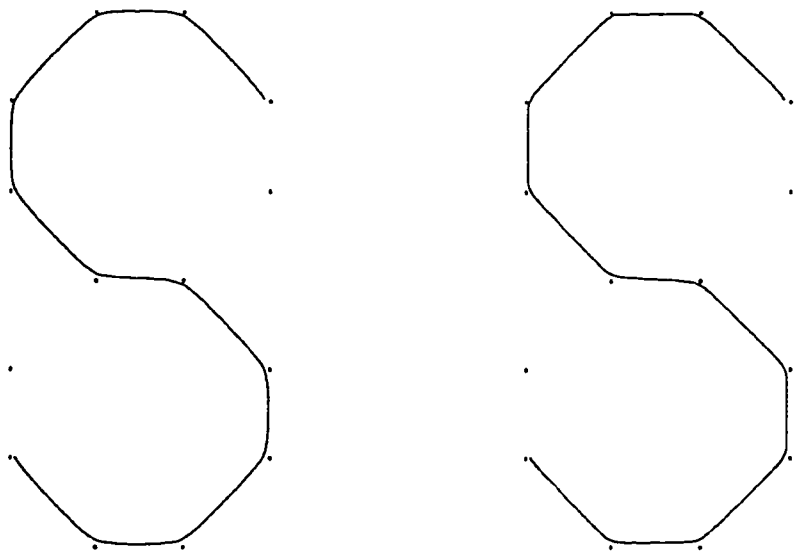

Figure 4.3: Cubic and Conic Spline Curves ($r_i = 5$).

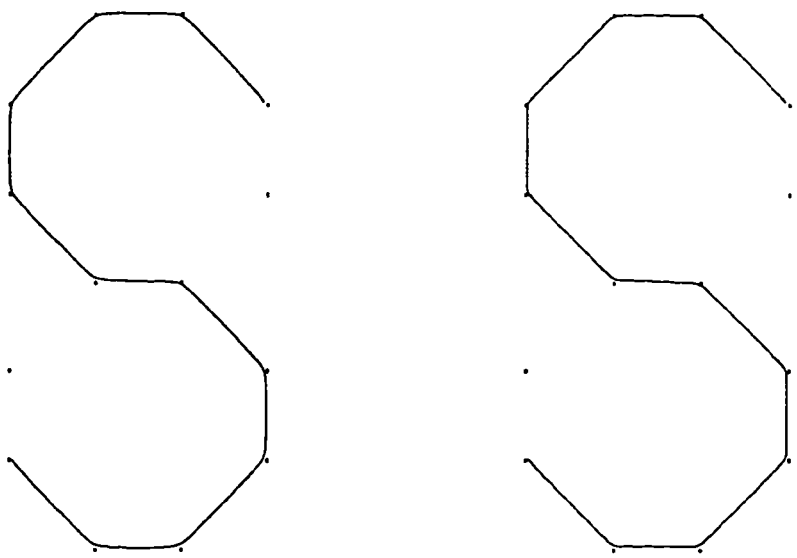

Figure 4.4: Cubic and Conic Spline Curves ($r_i = 10$).

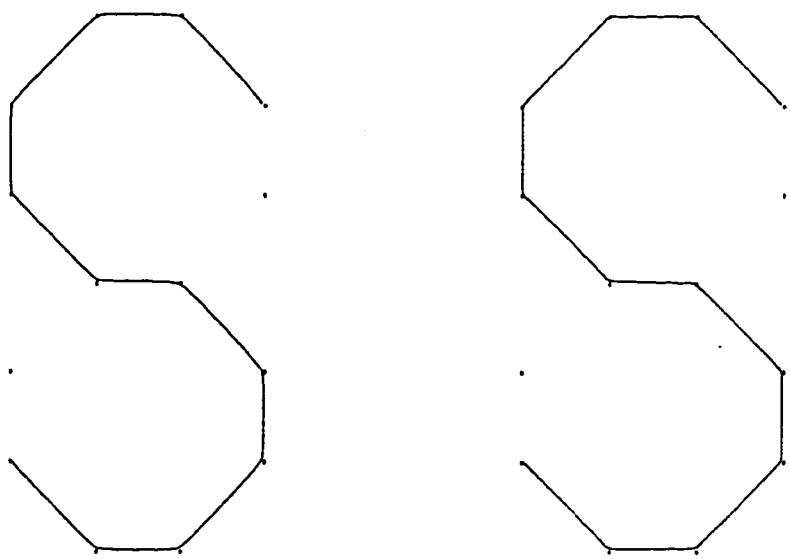

Figure 4.5: Cubic and Conic Spline Curves ($r_i = 15$).

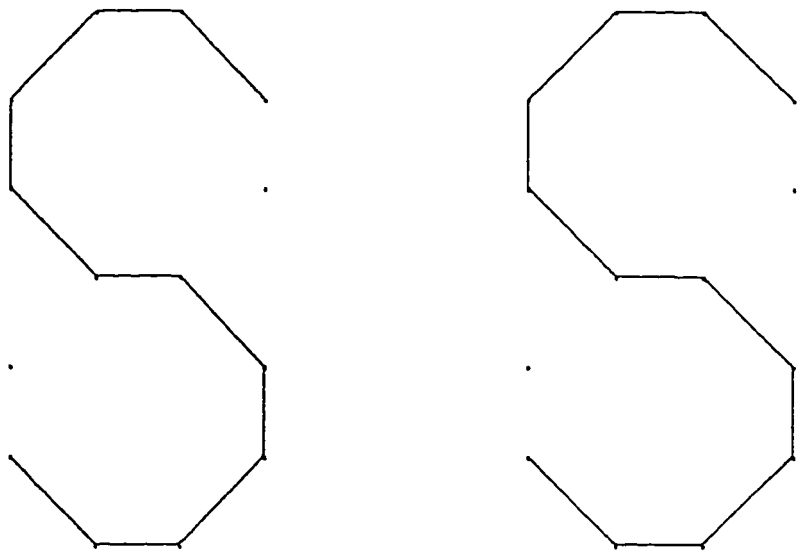

Figure 4.6: Cubic and Conic Spline Curves ($r_i = 50$).

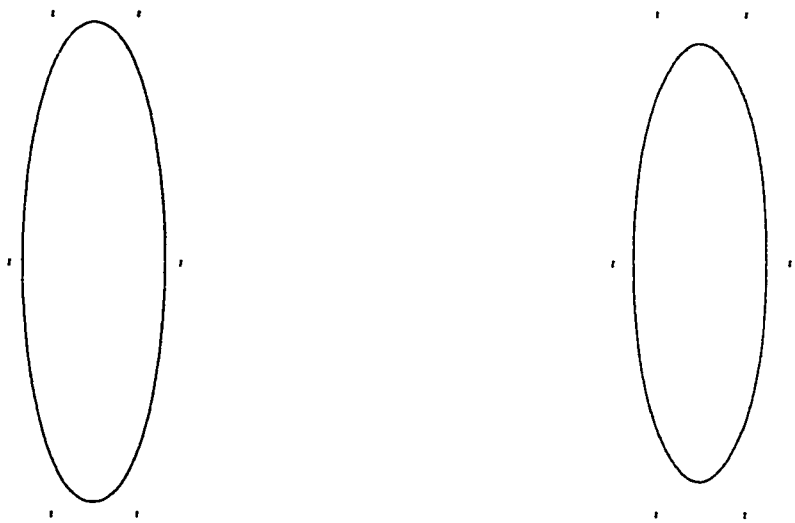

Figure 4.7: Cubic ($r_i = 3$) and Conic Spline Curves ($r_i = 2$).

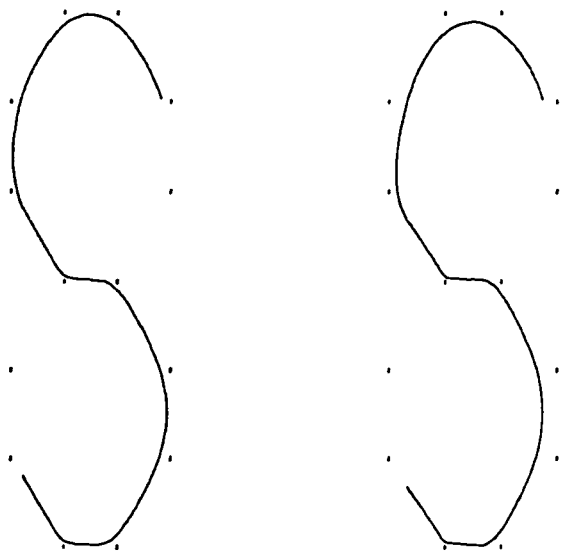

Figure 4.8: Effect of consecutive interval tension ($r_i = 10$) applied at two different intervals on cubic and quadratic (conic) spline curve.

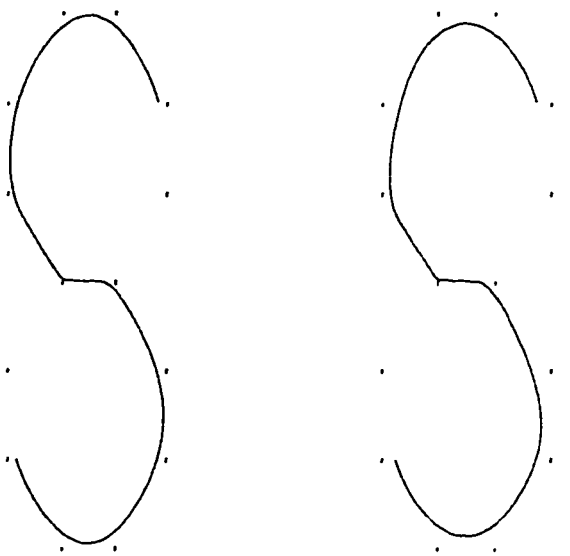

Figure 4.9: Effect of consecutive interval tension (r_i=10) in a particular region.

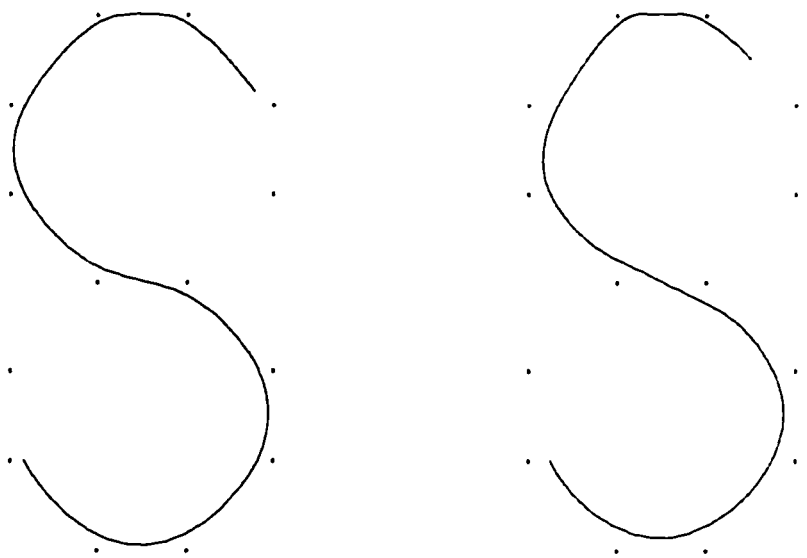

Figure 4.10: Effect of interval tension (r_i=5) in a particular region.

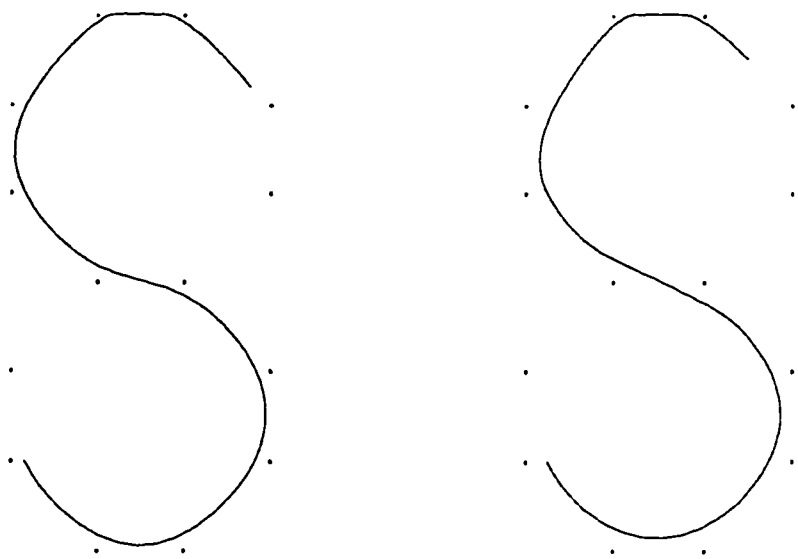

Figure 4.11: Effect of interval tension (r_i=10) in a particular region.

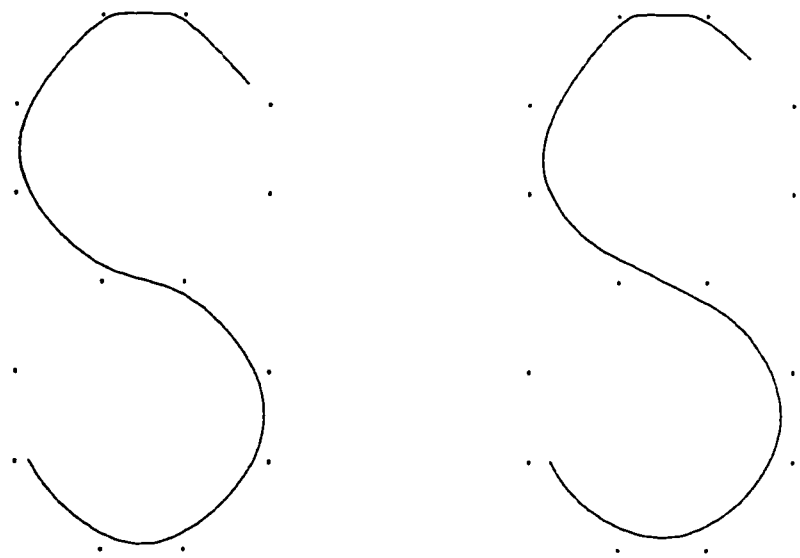

Figure 4.12: Effect of interval tension (r_i=15) in a particular region.

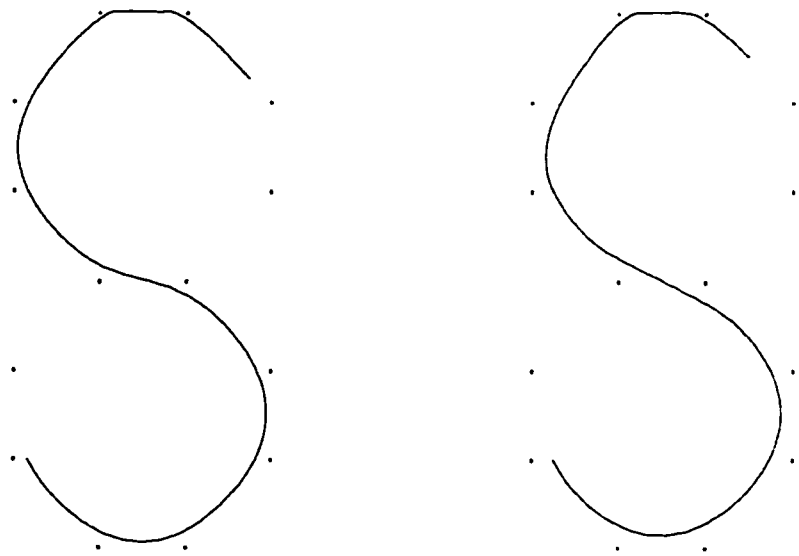

Figure 4.13: Effect of interval tension (r_i=50) in a particular region.

Figures 4.1 and 4.7 illustrate the rational cubic and rational quadratic (conic) spline curves (local tension). The left figure is plotted with the rational cubic B-spline method and the right figure is plotted with the rational quadratic (conic) B-spline method. At $r_i = 3$, a cubic curve is freeform, similar curve is obtained by quadratic (conic) with $r_i = 2$ (see figure 4.1 and 4.7) with slight loss in smoothness. Different curves are drawn illustrating the effect of increase in tension (see figure 4.2 to figure 4.5) and the effect of global tension is demonstrated in figure 4.6. The effect of increasing tension in two consecutive intervals generates more emphasis of tension at the joining point. (see figure 4.8 and 4.9). The effect of interval tension is demonstrated in figure 4.10 to 4.13.

4.3 Conclusion

C^1 continuous rational quadratic (conic) B-splines are obtained as an alternative to the rational (cubic) B-splines[7]. These quadratic (conic) B-splines do less computation. Compared to the rational cubic B-splines, the computation time taken by rational quadratic (conic) B-splines is less.

4.3.1 C^2 Freeform Spline Curves and Corresponding Surfaces (Bicubic)

The basis functions and their corresponding design curves in [7] is revisited, with respect to shape parameters in each interval. The method of computation is selected through the generation of the Bezier points from the control points, which makes the computations very convenient. Both the local and global shape effects can be

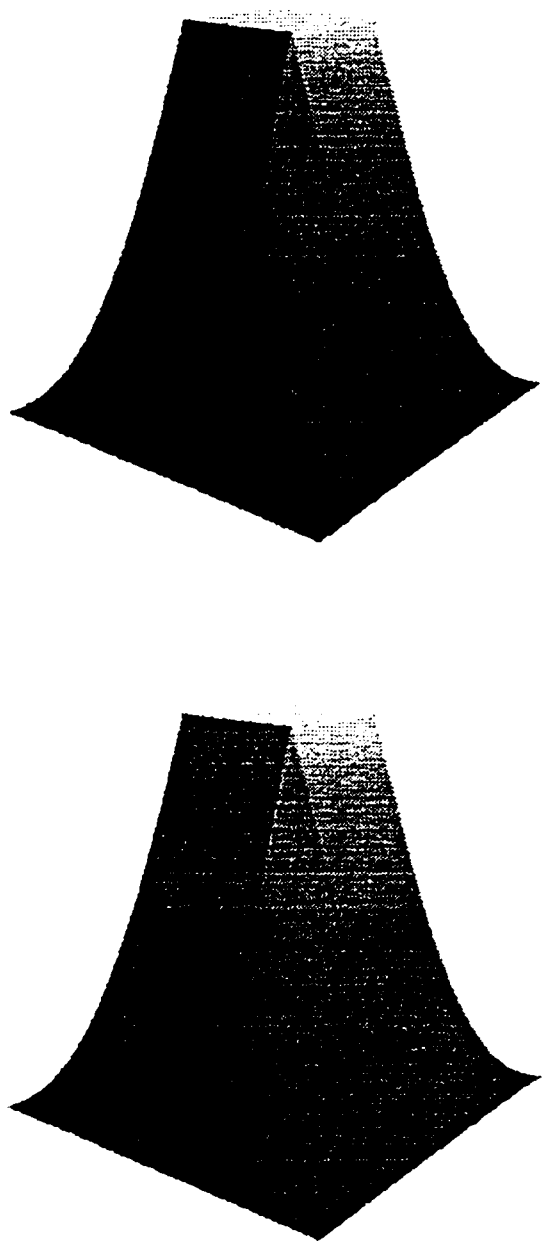

Figure 4.14: A Freeform Rational Bicubic and Biquadratic Surface (r_i=3).

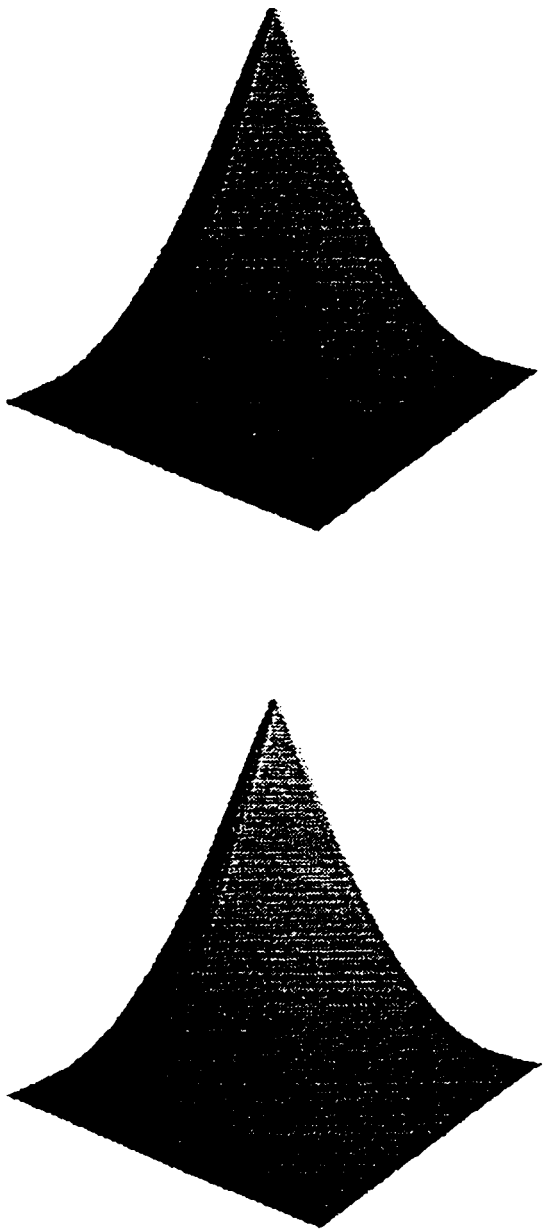

Figure 4.15: Effect of Global Tension on Bicubic and Biquadratic Surfaces (r_i=50).

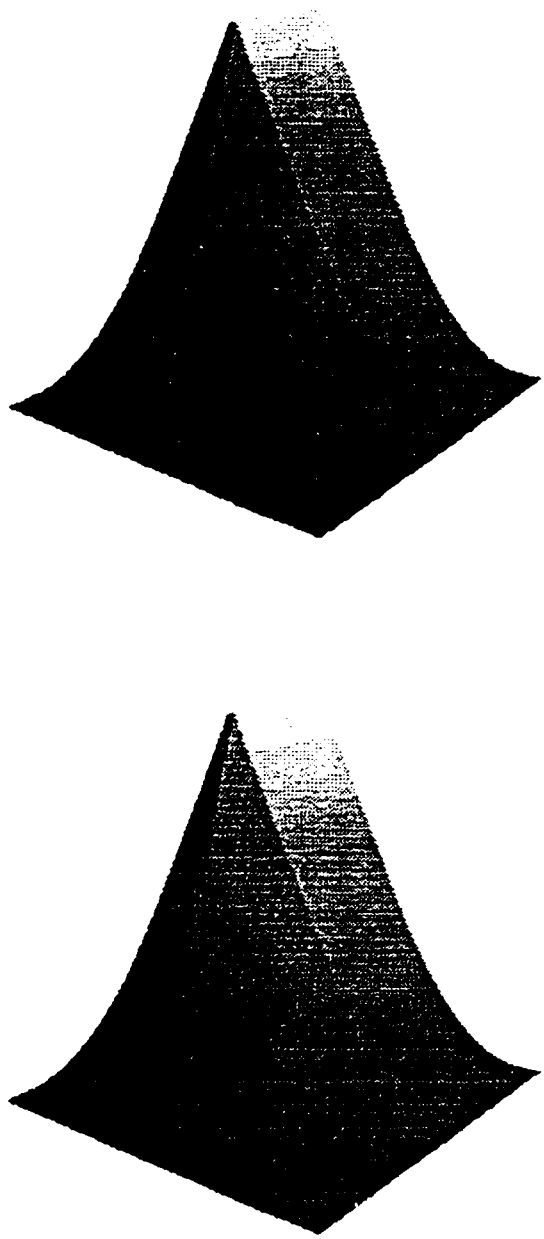

Figure 4.16: Effect of tension along t direction (r_i=50).

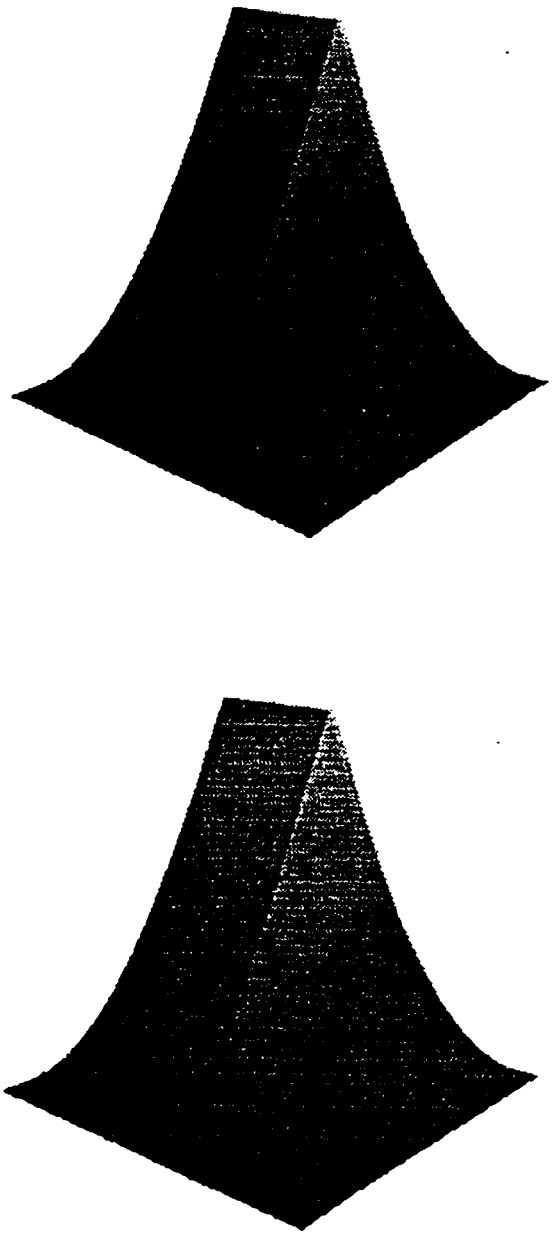

Figure 4.17: Effect of tension along $\tilde{t}$ direction (r_i=50).

achieved in a well controlled way.

The idea of the C^2 rational B-spline is generalized and extended to achieve a C^2 parametric rectangular freeform rational bicubic surface, which can be controlled locally and globally.

The surface has been designed through the sum of products of the bivariate rational B-spline basis functions [7]. This surface is designed such that, one shape parameter is associated with each interval. The shape parameters can be used both locally and globally, to tighten or loosen, the surface along and or across, the network of curves associated with each knot. Computation of the surface has been suggested through the Bernstien-Bezier representation, which is convenient.

4.3.2 C^1 Freeform Spline Curves and Surfaces (Biquadratic)

The idea of C^1 freeform curves and surfaces has been implemented to achieve corresponding C^1 freeform quadratic spline curves and surfaces. These curves satisfy all the properties of freeform curves, namely local and global tension. One shape parameter is associated with each interval, to control it locally and globally.

Figure 4.14 shows the effect of freeform surface (the figure on the top is drawn using Bicubic and the one below using Biquadratic). Similarly, figure 4.15 shows the effect of Global tension, figure 4.16 and 4.17 indicates the tension along t and $\tilde{t}$ directions.

4.3.3 Time Complexity Table

Theoretical Time Analysis: The Rational cubic B-spline functions are of degree three, it takes $O(n^3)$ time to compute those functions. Since there are two Rational

quadratic B-spline functions each of degree two, each of it takes $O(n^2)$ time. The total time taken overall is $O(n^2)$.

Type of Splines	Time Complexity
Rational Cubic B-Splines	$O(n^3)$
Rational Quadratic (Conic) B-Splines	$O(n^2)$

Table 4.1: Comparison of Time Complexity of Rational Cubic and Rational Quadratic B-Splines.

4.4 Future Work

- B-Spline formulation can as well be extended to interpolatory splines.
- Biased shape parameters can be incorporated, apart from point and interval tension effects.
- Can be used in visualization of data and designing of fonts.
- A recursive division (subdivison) algorithm can be developed for dividing the rational quadratic curve segments developed in [8].

Bibliography

[1] Jacques Bertin. Semiology of Graphics. *The Univ. of Wisconsin Press, Nadison*, 1993.

[2] Muhammad Sarfraz. A Rational Quadratic B-Spline Method with Shape Control. *Proceedings of the tenth International Symposium on Computer and Information Sciences*, 2, 1995.

[3] G. E. Farin. *Curves and Surfaces For Computer Aided Geometric Design*. Academic press, New York, 1988.

[4] I. J. Schoenberg. Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions. *Quart. Appl. Math,*, (4):45–99, 1946.

[5] Hearn and Baker. *Computer Graphics, Principles and Practice* . Addison Wesley, 2nd edition, 1994.

[6] Foley, Van Dam, and Hughes. *Computer Graphics, Principles and Practice.* Addison Wesley, 2nd edition, 1990.

[7] Muhammad Sarfraz. C^2 Rational B-Spline Surfaces with Tension Control. *New Advances in CAD & Computer Graphics*, 1, 1993.

[8] Muhammad Sarfraz. Scratchy Notes.

[9] C. Yao and J. Rokne. An Efficient Algorithm for Subdividing Linear Coons Surfaces. *Computer Aided Geometric Design*, (8):291–303, 1991.

[10] J. M. Lane and R. F. Riesenfeld. A Theoretical Development for the Computer Generation and Display of Parametric Surfaces. *IEEE Trans. Pattern Anal. Machine Intell.*, (2):35–46, 1980.

[11] J. M. Lane, L. C. Carpenter, and T. Whitted. Scan Line Methods for Displaying Parametrically Defined Surfaces. *Comm. ACM*, (23):23–34, 1980.

[12] E. Catmull. *A Subdivision Algorithm for Computer Display Of Curved Surfaces.* PhD thesis, Univ. of Utah., 1974.

[13] Muhammad Sarfraz. A Freeform Rational Cubic Spline with Shape Control: A Model For Computer Graphics. *J. Sc. Res.*,, XXI(1), 1992.

[14] N. Dyn, D. Levin, and J. Gregory. A 4-Point Interpolatory Subdivision Scheme for Curve Design. *Computer Aided Geometric Design*, (4):257–268, 1987.

[15] G. M. Chaiken. An Algorithm for High Speed Curve Generation. *Computer Graphics and Image Processing*, 3, 1974.

[16] D. V. H. Doo and M. Sabin. Behaviour of Recursive Division Surfaces Near Extraordinary Points. *Computer Aided Design*, 10, 1978.

[17] Alain Le Mehaute and Florencio I. Utreras. Convexity-Preserving Interpolatory Subdivision. *Computer Aided Geometric Design*, (11):17–37, 1994.

[18] Yu Yu Feng, Fa Lai Chen, and Hong Ling Zhou. The Invariance of Weak Convexity Conditions of B-nets with Respect to Subdivision. *Computer Aided Geometric Design*, (11):97–107, 1994.

[19] W. Boehm. Curvature Continuous Curves and Surfaces. *Computer Aided Geometric Design*, (2):313–323, 1985.

[20] W. Boehm. *Geometric Modeling*, chapter Smooth Curves and Surfaces in Farin, G.E., pages 175–184. eds. 1987.

[21] W. Boehm, G. Farin, and Kahmann. A Survey of Curve and Surface Methods in CAGD. *Computer Aided Geometric Design*, (7):1–60, 1984.

[22] J. A. Gregory and M. Sarfraz. A Rational Spline with Tension . *Computer Aided Geometric Design*, (7):1–13, 1990.

[23] M. Sarfraz. Rational Quadratic B-Spline Curves and Surfaces with Local and Point Tension. *Submitted for publication*, 1992.

[24] V. Pratt. Techniques for Conic Splines. *Computer Graphics, ACM SIGGRAPH*, 1985.

[25] T. Pavildis. Curve Fitting with Conic Splines. *ACM Transactions on Graphics*, 2, 1983.